Republic of Iraq

Ministry of Higher Education and Scientific Research

University of Babylon\ College of Information Technology

Developing an Adaptive SVD-based Distributed Ledger of an E-voting System

A Thesis

Submitted to the Council of the College of Information Technology, University of Babylon, in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology\ Software

By

Rihab Habeeb Sahib Naher

Supervised by

Prof. Dr. Eman S. Al-Shamery

2021 D.C. 1442 A.H.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

يا أَيُّهَا الَّذِينَ آمَنُوا اذْكُرُوا اللَّهَ ذِكْرًا كَثِيرًا ۝ وَسَبِّحُوهُ
بُكْرَةً وَأَصِيلا ۝ هُوَ الَّذِي يُصَلِّي عَلَيْكُمْ وَمَلائِكَتُهُ
لِيُخْرِجَكُمْ مِنَ الظُّلُمَاتِ إِلَى النُّورِ وَكَانَ بِالْمُؤْمِنِينَ رَحِيمًا
۝ تَحِيَّتُهُمْ يَوْمَ يَلْقَوْنَهُ سَلَامٌ وَأَعَدَّ لَهُمْ أَجْرًا كَرِيمًا ۝

صدق الله العلي العظيم

﴿سورة الاحزاب: الآية 41-44﴾

Supervisor Certification

I certify that this thesis entitled "***Developing an Adaptive SVD-based Distributed Ledger of an E-voting System***" was prepared under my supervision at the Department of Software / College of Information Technology/ University of Babylon, by **Rihab Habeeb Sahib** as partial fulfillment of the requirements for the degree of **Master in Information Technology.**

Signature:

Name: **Dr. Eman S. Al-Shamery**

Title: **Professor**

Date: / /2021

The Head of the Department Certification

In view of available recommendations, I forward this thesis entitled "***Developing an Adaptive SVD-based Distributed Ledger of an E-voting System***" for debate by the examination committee.

Signature:

Name: **Dr. Ahmed Saleem Abbas**

Title: **Assistant Professor**

Date: / /2021

Certification of the Examination Committee

We chairman and members of the examination committee, certify that we have studied the thesis entitled (**Developing an Adaptive SVD-based Distributed Ledger of an E-voting System**) presented by the student **Rihab Habeeb Sahib** and examined her in its content and what's related to it, and that in our opinion is adequate with ( ) standing as a Thesis for the **Master Degree in Information Technology**.

Signature:

Name: **Dr. Tawfiq A. Al-Asadi**
Title: **Professor.**
Date: / /2021
(Chairman)

Signature:

Name: **Dr. Wesam S. Bhaya**
Title: **Professor.**
Date: / /2021
(Member)

Signature:

Name: **Dr. Ali Fawzi Najm**
Title: **Assistant Professor**
Date: / /2021
(Member)

Signature:

Name: **Dr. Eman Salih Al-Shamery**
Title: **Professor.**
Date: / /2021
(Supervisor)

Approved by the Dean of the College of information technology, University of Babylon.

Signature:

Name: **Dr. Hussain Ateya Al-Khalidi**
Title: **Professor.**
Date: / /2021
(Dean of the College of Information Technology)

Dedication

To those who pursue their goals and never give up

Acknowledgments

First and foremost praise is to Almighty Allah. It is a pleasure to acknowledge my debt to many people involved in presenting this thesis: initially, I would like to express my sincere gratitude and appreciation to my supervisor Prof. Dr. Eman S. Al-Shamery for her invaluable guidance, supervision, and untiring efforts during my study and especially the period of this work.

Special thanks to my family, family in law, and my husband for their continuous support and encouragement during the period of my studies. Also special thanks to Dr. Abulkadhem and his wife Reyam for the great help they introduced. Finally, I thank my colleagues from the Science Course and the staff of the department of Software for the help they have introduced to me.

Abstract

An election is an important event that happens in many countries. Paper-based elections are considered time-consuming for tallying votes, cost of materials, untrusted and no privacy can be assured as the votes can be tampered with due to the ability of the election committee that controls the process. Electronic voting systems may count the results in less time, less cost of materials, may save the privacy of citizens, but still considered untrusted as votes can be manipulated.

The proposed system deals with a distributed ledger of massive size using the Singular Value Decomposition (SVD) technique as a data and dimensionality reduction tool, a transparency tool for matching the election results with the ledger that is constructed as soon as a block of votes is constructed that holds the immediate results. Also, The SVD technique is used as a tool that distributes the ledger by casting its outputs to separated network nodes. Each time a block of votes is created, a ledger for that block is also created and holds a copy of the results in another form by applying the processes of the SVD technique.

The system consists of a user and administrator platforms distributed among four phases. The first phase is the preprocessing phase which includes preparing the lists of candidates and voters, building the distributed network nodes for storing and processing data, and creating the general ledger that holds the results that are distributed by using cloud services. The second phase is the

confirmation phase where each citizen is allowed to vote based on the international ID card and cast a vote only once.

The third phase is the e-voting phase that includes the main stages of the proposed system, it includes adding transactions (votes) to the block, retrieving data from the distributed network nodes to obtain, validate and update the ledger using SVD.

The final phase is the result phase, which shows the election results as part of transparency to the voters successfully and matched with the final ledger that holds the results formed by SVD.

The system is applied on 100000 voters and 100 candidates in which each block has 20-100 transactions (votes). One hundred volunteers voted to test the system and check the results for different voters and different candidates randomly.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

µs	Microsecond
AES	Advanced Encryption Standard
BLOB	Binary Large Objects
BMD	Binary Matrix Decomposition
DB	Data Base
DLT	Distributed Ledger Technology
EA	Election Authority
ECC	Elliptic Curve Cryptography
EVM	Electronic Voting Machine
IP	Internet Protocol
MD5	Message-Digest Algorithm
ms	millisecond
NIST	National Institute of Standards and Technology
ns	Nanosecond
OTP	One Time Password
SHA	Secure Hash Algorithm
SQL	Structured Query Language
SVD	Singular Value Decomposition
URL	Uniform Resource Locator

XAMPP	(cross-platform), Apache HTTP Server, mySQL, PHP and Perl

List of Symbols

A	Binary matrix
m	Rows for the binary matrix
n	Columns for the binary matrix
U	Left singular matrix
$S= \sigma$	Singular value matrix
V	Right singular matrix
U^T	Left singular matrix transpose
$V^T = V'$	Right singular matrix transpose
r	The rank of the matrix
$\hat{U}$	Economic left singular matrix
$\hat{S}$	Economic singular value matrix
$\tilde{U}$	Truncated left singular matrix
$\check{S}$	Truncated singular value matrix
$\tilde{V}^T$	Truncated right singular matrix
k	Lowest suitable rank
τ	Tolerance
V_n	Voter
C_n	Candidate

Chapter One

General Introduction

1.1. Introduction

Elections should bring democracy to countries. Their role is important for the future of citizens' life in all countries around the world. Elections have to be trustworthy and ensure the security of privacy for each voter. Electronic voting is the use of electronic devices to cast votes. E-voting may use voting machines or computers connected to the Internet. Trusting the e-voting system is important when it comes to the actual use of the electronic system within the political process [1].

Additionally, for tallying votes, there should not be too much time, as spending a long time waiting for results increase concerns about tampering with the results, as the entire voting process should be trusted and transparent [2]. One of the solutions to ensure that each vote is counted correctly and neither the election committee stores the database nor change votes (no third party can tamper any vote) where only eligible people are allowed to vote is distributed ledger technology. Electronic voting systems are not free from being tampered and voters cannot ensure that their votes were counted, also, announcing the results is a slow process because ballots must be gathered from different places and then counted by a single central institution [3].

The most frequent issue in elections whether they are paper-based elections or electronic voting systems is the problem of security, data manipulation, trust, and transparency. The transparency, assurance, and confidentiality issues of an e-voting system can at some degree of security be managed using Distributed Ledger Technology (DLT) in the

proposed system, in which the election results are distributed among several nodes in another form [4].

Many problems are having a third party controlling applications, transactions, elections, or whatever human needs. The initial use of DLT was focused on financial services especially cryptocurrencies but with time it is realized that DLTs can be used in Voting, Healthcare, Insurance, IoT, Supply chain, etc…. Opposition parties in different countries regularly claim that the election was tempered. Mainly two methods are used in voting, Electronic Voting Machine (EVM) and Ballot paper. Both these systems are not fully trustable. Another solution is online voting, but in a voting system, we need security, privacy, and anonymity of voters, and transparency in the counting of votes which is not possible with the existing technology. DLT could support these features to make democracy stronger [5].

To authenticate a voter, several technical solutions are needed to implement an electronic voting system, ballot secrecy, and security [6]. One of these solutions is using DLT and making it more powerful using a data mining technique, such as singular value decomposition (SVD).

The Singular Value Decomposition (SVD) in linear algebra factorizes a matrix into three matrices and has a wide range of applications in data mining, computer vision, statistics, and machine learning [7].

The singular value decomposition technique is a way to analyze a matrix that results in a low-dimensional representation of a high-

dimensional matrix. The SVD technique is mostly used to erase less important parts and produce an approximate representation with any number of dimensions that are desired where data is of a binary form containing data of zeros and ones [8].

This thesis proposes an online (web application) e-voting system of a distributed ledger aiming to make the ledger uncontrollable by one party depending on the singular value decomposition (SVD) technique that separates the ledger into parts, each part is represented as a matrix that is distributed to a single server in a cloud. Also, saving the privacy of the voter by saving his/her vote in a block that is secured by a hash function. Authentication is done by identifying the correct information for each citizen. Transparency, by adding the ability to show the results during the election event till the end of time, and integrity is reached by matching the online election results with the copy of results that are formed by SVD for each block of votes. The proposed system is easy to use by voters and tries to make it as trusted as possible by saving copies of results that are in a distributed manner during the election event.

1.2. Problem Statement

An acceptable trusted and secured e-voting system for election is a requirement for all citizens around the world. Results can be manipulated which is the main issue. Also, transparency is needed for showing the results in real-time which makes such systems more trusted.

Anonymity is an issue that is needed to keep voters anonymous and safe for voting for the desired candidate. These main issues should be considered to develop an online e-voting system that covers these

features. The proposed system should keep a copy of instant results in an encoded form from the beginning till the end of the election event, along with showing the results and keeping citizens anonymous adding a degree of security as high as possible to reduce possible chances of different attacks.

1.3. The Thesis Challenges

The proposed e-voting system faced several challenges, as follows:

1- The time wasted for confirmation and validating the ledger should be as little as possible (in nano or microseconds).

2- Employing a mechanism to rate the transparency of the election event in terms of presentation and accuracy of results.

3- The interaction of a dynamic real-time system with real-world entities.

4- Adding the results in the ledger and all following processes should be performed within fast response and less amount of time, due the time set for the whole election event.

5- Collision avoidance may happen to the system, due to the large number of votes that flow to the systems all at once, which may cause the system to stop.

1.4. The Aim of Thesis

This thesis aims to achieve the following:

1- Developing an online (web application) e-voting system based on an adaptive SVD technique that is used as follows:

- Using the SVD technique as a data and dimensionality reduction tool without losing important information.
- Using the SVD technique as a distributed tool to save the election result in a distributed manner in separated network nodes.
- Using the SVD technique as a matching tool, which forms incremental ledgers holding the results for each block, which is then matched with the online election results.

2- Producing another copy of the results in an encoded form represented by an incremental ledger.

3- Run the e-voting system dynamically in real-time with no control over the system.

4-The proposed e-voting system aims to reduce manipulation in results as much as possible by achieving:

- Transparency: showing the results in real-time which leads to being a desired feature for voters.
- Integrity: a degree of integrity should be achieved by saving an immediate copy of the results by applying SVD on every block of transactions (votes) and distributing the results in another form to distributed network nodes (separated servers).
- Anonymity: a voter would use an ID created for him/her that is then added as a transaction (vote) in a block, and this block is encrypted using the hash value (SHA256).
- Uniqueness: every voter has the right to vote only once and is not allowed to enter the voting phase. But, can view the results.

- Verification: verification should be performed for the ledger of results that should contain a value of 1 for each voter (represented in a column corresponding to the desired row of the candidate).
- Security: several degrees of security should be applied by saving the votes in an encrypted block using the hash function SHA256, other security features are employed within the cloud services by encrypting all information (databases, data, files, folders, and servers).
- Throughput time: Reducing the time needed for performing all processes as little as possible (in nano to milliseconds).

1.5. The Thesis Contributions

This thesis leverage SVD as a base tool to propose an electronic voting system making the following contributions:

1- Using an Adaptive SVD as a means to reduce the dimensionality of the ledger without losing information.

2- Distribute the ledger of results in real-time as soon as the system begins to run by applying SVD and cast the results of the ledger in separated network nodes.

1.6. Related Work

Yasmine 2013 [6], proposed an internet voting system in Iraq. Before the date of the election begins, the voters should be registered in the database. A 10 digits unique password is generated for each voter. On Election Day, the voters' passwords and information are entered on the registration form. If the information is accepted; the voters enter the page where they can select the desired candidate to vote for, else the user

cannot vote. Only an eligible voter can cast a vote once and the whole process is controlled by the administrators of the central organization.

In this thesis, the proposed system implements the system without the need for advance registration for each voter, instead, the voter will enter his\her international ID during the day of the election begins, the database will include the International ID, name, and the city for each citizen. Then preceding the process of voting only once. Also, the file (ledger) that contains the votes is distributed among several nodes to ensure that the system is decentralized and no particular party controls the ledger using the SVD data mining technique aiming to use it as a matching tool to check the transparency of the results and as a dimensionality reduction tool for saving time.

Aakash, et al., 2020 [9], proposed an online voting system in which the back-end server takes care of authenticating the users and maintaining necessary details using the XAMPP (Apache server). The user interface at the server's end enables creating the election on behalf of the users. Also, the admin can log in to his account and can manage the whole voting process by adding a new election, generating id for the user, verifying the users, and generating results.

In this thesis, the proposed system handles the whole system dynamically in which the administrator starts the system through Azure with several servers to distribute the ledger of results in separated network nodes and test the transparency of the system by having another copy of the results gathered through blocks of votes to be finally matched with the online election result.

G. Kalaiyarasi 2020, [10] proposed an e-voting system using an Android platform that allows the voter to cast the vote with no need to reach the polling booth. The application has an authentication process to avoid any attempt of fraud using the one-time password (OTP) that will be given to the voter after he\she login by giving the name, ID, and location to vote. The voter must enter the number of mobile for generating the OTP that will be submitted to consider the vote as valid. The firebase server is used to authenticate the number of mobiles and sends back an OTP to the voter. All the counts for the casted votes are encrypted using the Advanced Encryption Standard 256 (AES256) algorithm and saved in the database to avoid any control by other than the administrator that will decrypt the results then announce them.

In this thesis, the proposed system is a web application that is suitable for all citizens using any device connected to the internet. The results can be viewed at any time and a certain number of votes are structured in blocks secured by the hash function SHA 256. Also, the whole system has its data and files encrypted using the Azure encryption service and firewall as a service that secures the network. Also, the proposed e-voting system can check the results whether are manipulated or not by employing SVD as a matching tool that checks the transparency of the election event by converting the results immediately into another form distributed in separated places.

Ahmed 2017 [11], proposed a design for an electronic voting system that mainly tries to cover four requirements, authentication where only eligible citizens are allowed to vote, anonymity where the voter has

to remain anonymous, accuracy in which every vote is unique and counted, and the verifiability to make sure all votes are counted correctly.

In his proposed system, the candidate will be the first transaction added to the block, containing the name of the candidate and considered the base block, in which votes for that candidate are placed on top of the base node and will not be counted as a vote. Each vote will get recorded and the blockchain will be updated. Each vote is sent to a specific candidate's node, and the nodes then add the vote to the blockchain. To ensure decentralization, the voting system will have a node in each area where the election is held. Limitations in this system are the ability to change and cast a vote by a hacker using malicious software already installed on the voter's device.

In this thesis, the proposed system tries to implement the mentioned requirements where only eligible citizens are allowed to vote depending on the valid ID saved in the database that will return the name of the citizen ensuring his\her personal information, the voter remains anonymous as the whole block containing the IDs of voters is hashed using SHA256, every vote is unique and counted in a ledger that will be distributed by using SVD technique, and make sure all votes are counted correctly by employing this technique.

Ramya 2020 [12], proposed an online e-voting system in which casting a ballot and saving information is done using a cloud server for saving time and voting from any place by using an online web-based voting system. By using a cloud there is the capacity to offer to cast a

ballot platform to voters in the country and outside through a web-based casting a ballot. The voter uses the username and secret key for entering the system and casting a ballot, the results can be simply viewed after the end date.

In this thesis, the proposed system uses cloud services to save a database that contains citizens' information and files reducing the size of the general ledger we are dealing with, at the same time speeding up the search needed for each voter to enter the system using his\her information, which is implemented by the cloud engine. Also, using the cloud services to deal with the distributed file based on the singular value decomposition (SVD).

Sekar 2020 [13], proposed a decentralized e-voting system using blockchain for elections to provide transparency and flexibility. The system consists of three modules, the user validation model that uses biometric (fingerprint) and other information (name, gender, address) to verify the user. At the time of voting, all this information is hashed using the Message-Digest algorithm (MD5) and checked with the election database.

The second model is the dynamic ballot loading model, the voters will have to go to the nearest polling booth. The third model is the Acknowledgment model after casting their vote it is given as acknowledgment to the user. The Election Authority (EA) has the role of creating a vote, not allowing more than one vote for each voter. The results are published after tallying the votes which are made by EA.

In this thesis, the proposed system distributes the ledger with the help of cloud services and the SVD technique. No need for the voter to go to the nearest ballot station because the system is an online web application that can be reached by any device. Since, using blockchain can lead to some drawbacks like scalability and latency, the proposed system leverage the features of a distributed ledger with security features offered by cloud services and employs the SVD technique as a tool used for ensuring transparency in real-time.

Table (1-1) shows the criteria of e-voting systems features for the related work and the proposed e-voting system, each has its implementations that achieve the desired feature. Although, some features can have a lack of performance due to challenges that may face the systems such as, it is almost impossible to guarantee that any system can be fully secured due to the development of malicious software and technology. Additionally, table (1-2) illustrates the mechanisms, techniques, and functions used for each e-voting system including the proposed e-voting system.

Table (1-1). Criteria of e-voting systems for the related works and the proposed system

	Criteria features for the e-voting systems					
Authors	**Transparency**	**Real time**	**Trusted**	**Secured**	**Authentication**	**Anonymity**
[7] Yasmine M. Tabra (2013)	✗	✗	✗	✗	✓	✓
[10] Aakash (2020)	✗	✗	✗	✗	✓	✓

[11] G.Kalaiyarasi (2020)	✗	√	✗	√	√	√
[12] Ahmed (2017)	√	√	√	√	√	√
[13] Ramya (2020)	√	✗	√	√	√	√
[14] Sekar (2020)	√	✗	√	√	√	√
The proposed system	√	√	√	√	√	√

	Technologies, techniques, and functions for the e-voting systems							
Authors	**Smart phone App.**	**Web App.**	**Blockchain Tech.**	**DLT Tech.**	**Data Mining Tech.**	**Hash Function**	**Cryptography Functions**	**Cloud services and servers**
[7] Yasmine M. Tabra (2013)	✗	√	✗	✗	✗	✗	✗	✗
[10] Aakash (2020)	✗	√	✗	✗	✗	✗	✗	√
[11] G.Kalaiyarasi (2020)	√	✗	✗	✗	✗	✗	√	√
[12]Ahmed (2017)	✗	√	√	√	✗	√	✗	✗
[13] Ramya (2020)	✗	√	✗	✗	✗	✗	√	√
[14]	✗	✗	√	√	✗	√	✗	✗

Sekar (2020)								
The proposed system	X	√	X	√	√	√	√	√

Table (1-2). Technologies, techniques, and functions for the e-voting systems for the related works and the proposed system

1.7. Thesis Outline

This thesis is organized as in the following:

Chapter two: Introduces general theoretical concepts for the e-voting system, Azure cloud services, DLT, Hash functions, SVD, and the settings needed for the system.

Chapter three: Introduces the developed e-voting system.

Chapter four: Introduces the experimental results of the developed e-voting system.

Chapter five: Summarizes the conclusions and suggested works for the future.

Chapter Two

Theoretical Fundamentals

2.1. Introduction

This chapter presents the concepts that are in concern to the proposed e-voting systems. The first part explains the development of elections from paper-based voting to electronic voting systems. It also presents the features needed in every e-voting system to be as efficient as possible.

The second part will present all techniques and mechanisms used sequentially in the proposed system, starting with Azure cloud services that are used to employ servers, storage, and some security services, distributed ledger technology (DLT) and its applications, Hash functions, and the singular value decomposition (SVD), and how it works. Also, presenting the settings that are needed for the proposed system.

2.2. Development of Voting for Elections

Voting is a way to introduce democracy and though there are many measures to add complex security, it is not free from frauds, attacks, and manipulation [14]. There exist many different types of e-voting systems, some are still paper-based and voters must attend polling stations [15]. Using paper ballots and tallying votes not only lead to errors but also is a time-consuming process [16].

Digital or electronic voting is the use of electronic devices to cast votes through an internet browser or voting machines. These are often referred to as e-voting when voting using a machine in a polling station, and I-voting when using an internet browser, though they are used

interchangeably [17]. Figure (2-1) illustrates the development of voting for elections till today.

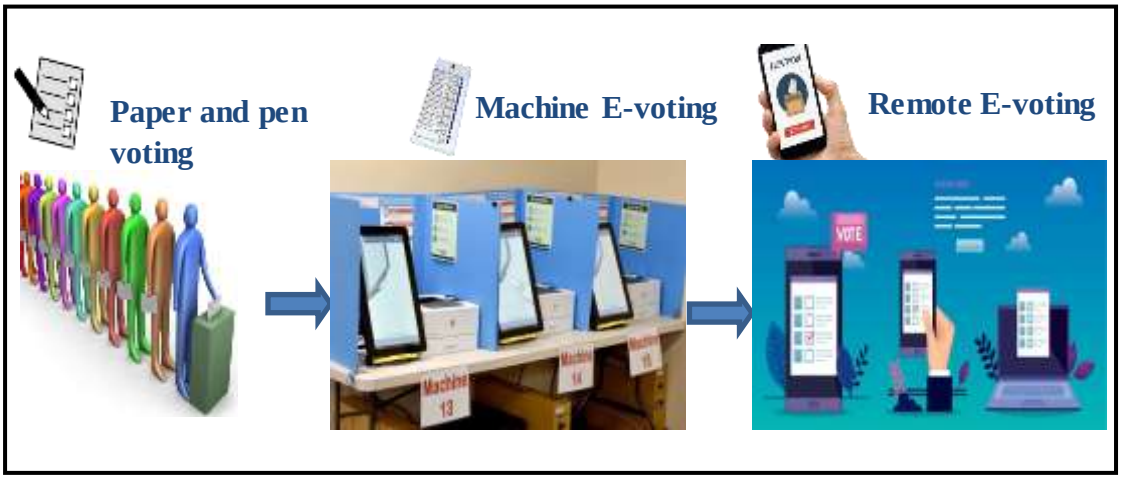


Figure (2-1). Development of voting for election events

Compared to paper-based voting, electronic voting systems are more economic systems that address transparency and privacy. Because e-voting systems essentially rely on the internet platform, the critical challenge facing E-voting is the security risks. To reduce such risks, many protocols have been proposed widely related to the privacy of ballots and eligibility by keeping the identities unknown using cryptography techniques, uniqueness by adding biometric information that is then encrypted leading to a robust universal and individual verifiability as much as possible [18].

However, the risks of e-voting are so essential, that has prevented many governments from implementing it. If any intervention occurs with an e-voting system, the possible costs are fatal. The existing voting systems, whether they are electronic or traditional, do not include sufficient levels of transparency. In either case, it is difficult for voters to

ensure that their electoral votes are tabulated carefully and accurately by the central election office[16].

No matter how many positive features e-voting systems achieve, perfection is an unattainable goal. However, the attempt of improvements till today is research attention for many researchers [19].

Internet voting can greatly reduce opportunity costs as it allows citizens to vote using their own devices such as computers or mobiles. In that concern, E-voting cannot only be a viable option to vote but the most efficient one [16].

Nowadays e-voting systems are one of the most known issues of e-democracy, which has led to developing applications and security techniques for such necessity [20].

2.3. Types of E-voting Systems

There are two main types of e-voting systems, each with its benefits and drawbacks, mentioned as following:

2.3.1. Machine-Based E-voting Systems

Compared to paper-based voting, electronic voting machines have become effective. The aim is to apply security and to defeat the limitations that arise in traditional ballot systems. The machine-based e-voting systems can include two types based on the technology used as follow:

I. Biometric machine-based e-voting system

This type of voting system usually uses biometric information technology for identification, the information of voters with their biometrics (fingerprints, DNA, eye retinas, facial patterns, etc.) are stored within the database. Then the effect of their biometrics is verified by the controlling unit. The microcontroller of these machines compares the voters' statistics with the present information stored at some stage in the registration of the voter. Voters are allowed to cast their votes if the statistics match with the already existing statistics. These systems do not allow duplication of votes and the results are announced as early as possible [21].

However, In past years, many electronic systems for elections failed because security is not guaranteed to the privacy protection of a vote, particularly in the cases of brute force attacks and in the long-medium term[22].

II. Blockchain-based e-voting systems

Transparency for viewing the results and security are the most frequent issues in elections. The technology of blockchain is one of the solutions that is employed to reduce several problems in voting events. A voting system based on blockchain could decentralize control where no organization can control the system by itself [23].

Blockchain has the property of being distributed, unchangeable, and is a transparent ledger that cannot decline the truth. Blocks of transactions (data) are linked to each other in sequence by hash

functions, the previous hash value is used in the next block, and so on. Any attempt to alter information in a single block is a difficult process, as the link to the following blocks should be also altered. The database is public and accessed by all the users in the network. If the database was owned by a hacker who attempts to attack the network, then the database owned by users will be different, and according to the consensus mechanism that is agreed by the most number of users, the database of the attacker will be denied [24]. Figure (2-2) shows a blockchain e-voting scheme.

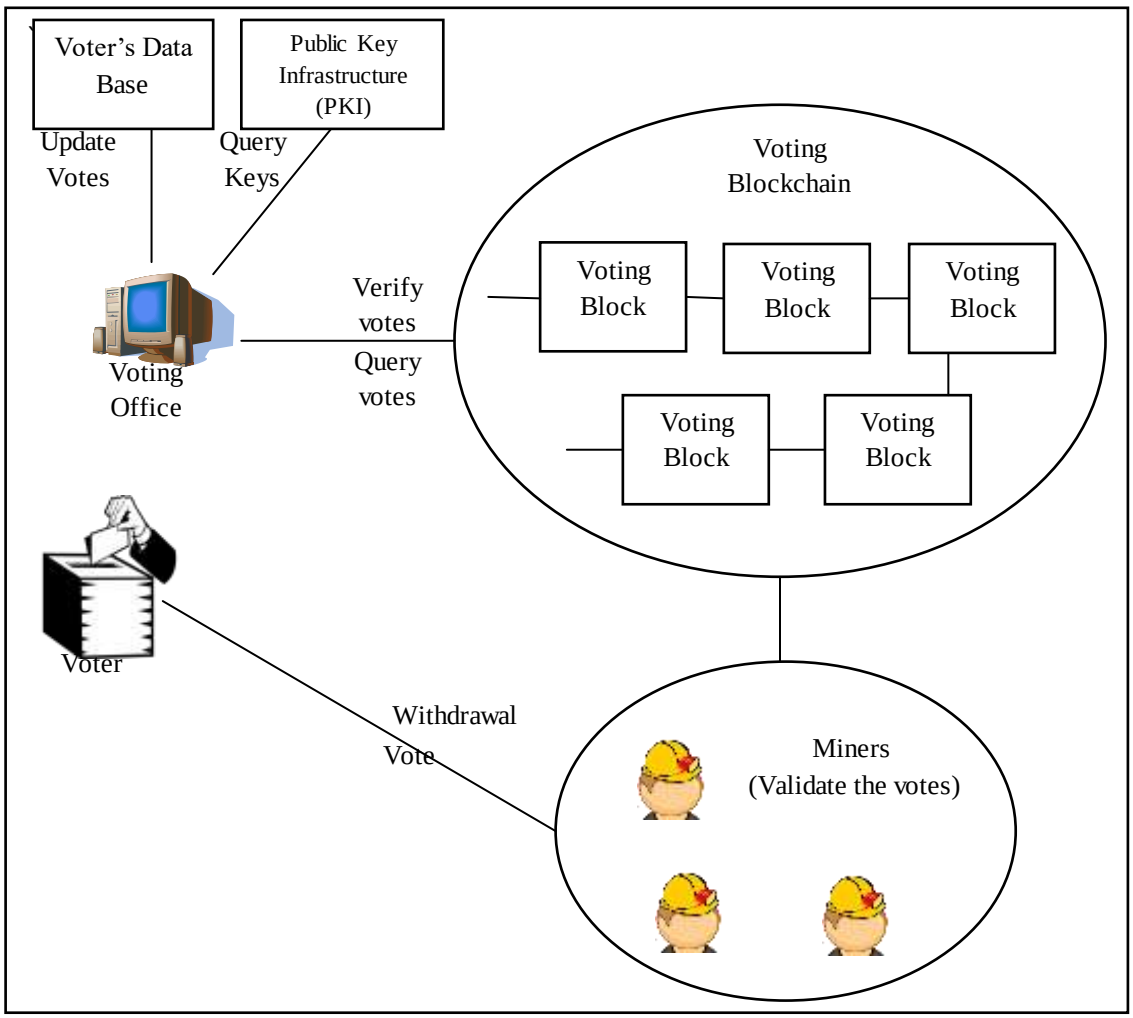


Figure (2-2). Blockchain-based e-voting scheme[25].

Blockchain has the characteristics of security, immutability, transparency in voting systems. However, most implementations are for small elections, small communities and in most cases, it is for boardroom elections [26]. Also, there are some vulnerabilities, the most known one is the scalability problem in e-voting based-blockchain for dealing with millions of votes which is a security concern in blockchain [27]. Blockchain scales up every time a change occurs. In December 2017, Bitcoin reached a price of nearly 20,000$, the number of citizens around the world that are attempting to sell and buy cryptocurrency made transactions very slow. If this occurs with voting, it causes serious issues because results of elections should appear at most in hours, which is hard to scale[28].

There is a statistic on Bitcoin that can manage transactions for 7 seconds only. In an election, this can be done for thousands of people that can vote, but not millions[29]. For example, in the Brexit plebiscite, 35 million votes are cast, if those transactions are verified at 7 seconds, it would take 55 days to get everything done. So the response time and size of data size are a challenge. Other drawbacks can be securing information for personal users, new malware, or a private security key that may be lost. However, there are risks, drawbacks, and threats associated with benefits, it is important to notice that blockchain-based E-voting is still in its beginning. Hence, blockchain may take years to be an effective form for elections [27].

Whether it is a traditional e-voting system or based on blockchain, these types of systems using a voting machine require citizens to visit polling centers that may need distances to go, and due to the world

pandemic of COVID-19, such systems using voting machines are not suggested for future elections.

2.3.2. Internet-Based E-Voting Systems

The evolution in web technologies gave growth to a new application that will make the voting process very easy to use with more efficient skills. E-voting through the internet helps a voter cast and count the votes by using any device (computer or mobiles) without visiting the polling booth[10]. We break the internet-based voting systems into two kinds based on the technology or mechanism used as follow:

I. Ordinary online voting systems

The online voting system allows citizens to cast a vote through a non-controlled environment using smartphones and computers to access an e-voting website. To add a degree of security, such systems use a variety of mechanisms for identification or authentication such as the OTP (one-time-password) process that is used to show the difference between an automated bot and a human using a web service, to secure the website against spam-bot attacks [30]. Whenever a user enters the system and performs some action, a random number of 6 Digit Number is assigned to the phone of the user [31].

The use of OTP that is sent to the voter after voting for validation is a more secure method than one-factor authentication, which is why it is used in online transactions [32]. Figure (2-3) is an example of an e-voting system with OTP.

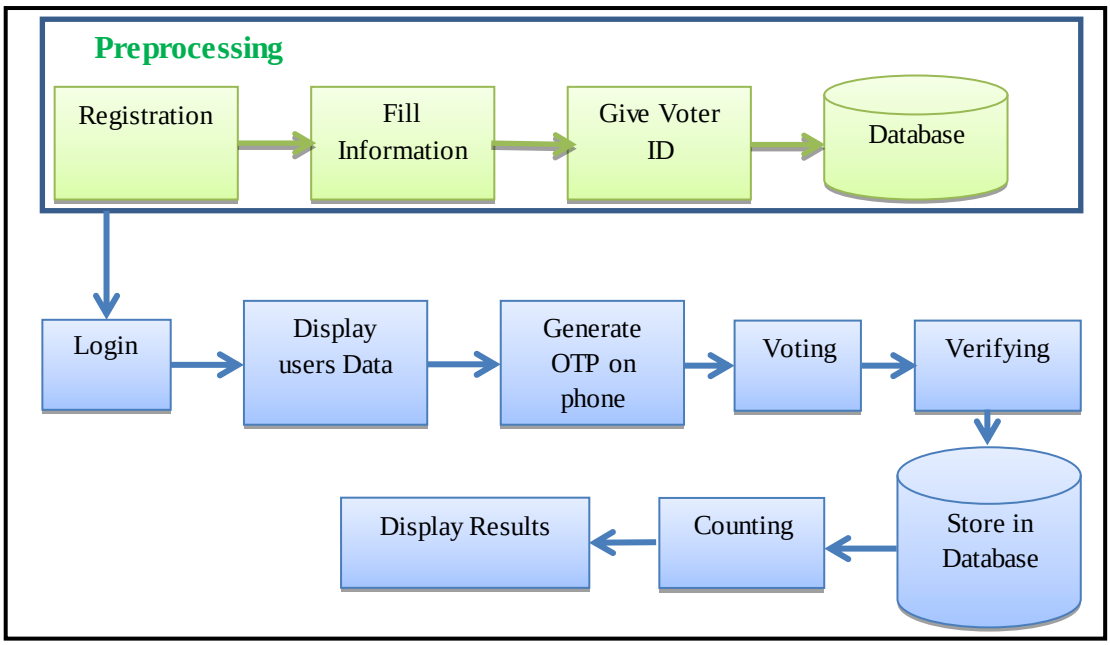


Figure (2-3). One Time Password (OTP) in e-voting system [31].

Other e-voting systems may use email for authentication but such an approach is not considered due to that emails are easy attacked, the Elliptic Curve Cryptography (ECC) algorithm is chosen as a good method as it has a key of small size which makes it more desirable than other public-key cryptographies, and keep users' anonymity by its homomorphic encryption property which encrypts the voters vote to increase the security of the system [33].

II. Blockchain or DLT E-voting systems

The use of devices such as mobiles or computers to access websites for e-voting systems based on blockchain technology (type of DLT) can be used, in which erasing or altering data in a blockchain is almost impossible [34]. It is a decentralized technology that can avoid a point of failure with the group working to confirm new legitimate transactions together [35].

Blockchain technology in e-voting systems is used as a database to save and cast votes securely by preventing any kind of tampering, in order to verify the eligibility while giving the voters access and permission to cast their votes using their devices from any location. This process increases voters' trust and offer better transparency towards the voting process [36]. Figure (2-4) illustrates blockchain with an e-voting system.

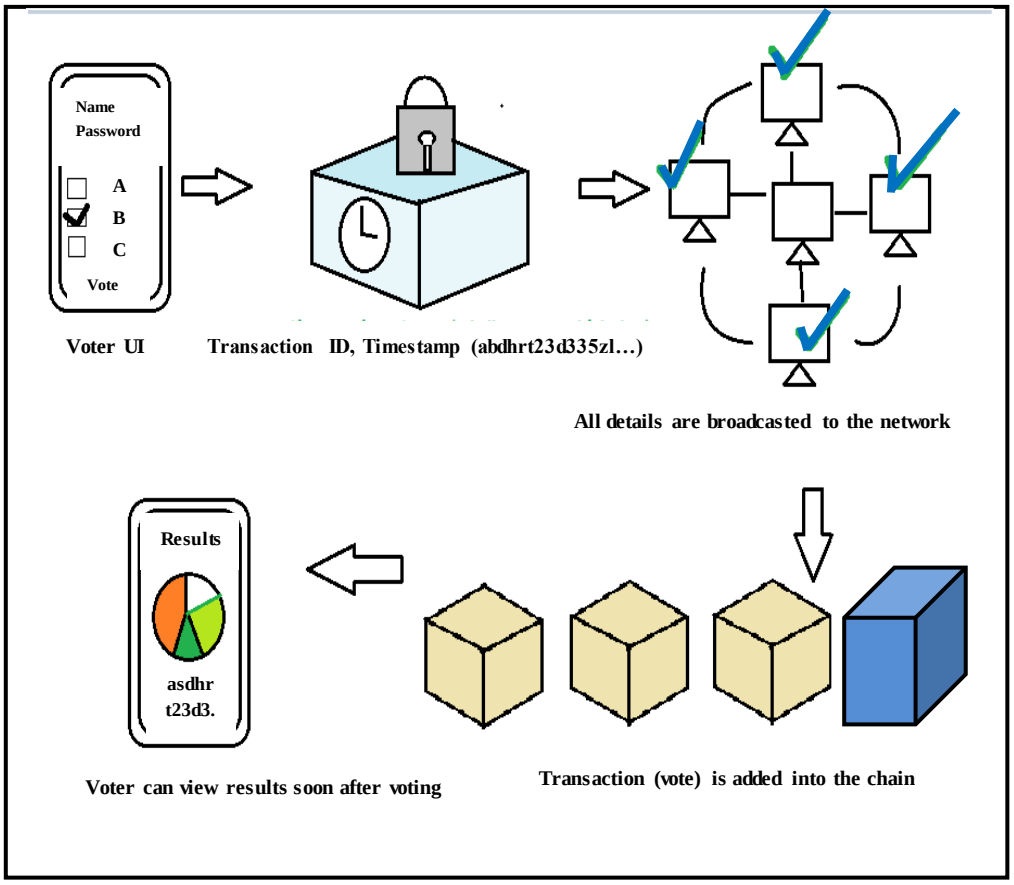


Figure (2-4). Blockchain e-voting system[37].

However, e-voting systems based on blockchain are still under research studies, as a future view to deal with a massive size of the ledger of the results. Also, the population of countries may affect the computing process of blockchain [38].

Many research gaps in e-voting have been presented that should be considered for future studies such as the use of untrusted systems, scalability attacks are additional disadvantages and should be resolved. Since the e-voting systems that are based on blockchain require further testing, the risks with the security issue and scalability for e-voting systems are not completely aware and blockchain-based voting can bring unknown security vulnerabilities and risks [39].

Issues such as skills for management and more sophisticated design are required for blockchain systems. For this reason, e-voting systems based on the blockchain should be applied initially to small regions. That is why blockchain-based technology is still at an early stage in an e-voting solution [40].

2.4. Distributed Ledger Technology (DLT)

The rise in technology nowadays helps citizens to exchange money through the internet without the need for physical transfer, only updates of database entries are required which is controlled by central authorities mainly banks. Also, for proper settlement of transactions between two unknown parties, a trusted third party is needed and this third party resolves the issue between both the parties in case of conflict. Many problems are having a third party for settling the transactions. Firstly, these third parties have to be paid for the services they offer. Secondly, it takes time for the transactions to settle e.g. for credit card transactions it takes three to seven days for the transaction to be completed. Finally, a third party can invalidate any transaction at any

moment to serve its purpose [5]. Figure (2-5) shows the centralization and decentralization ledger.

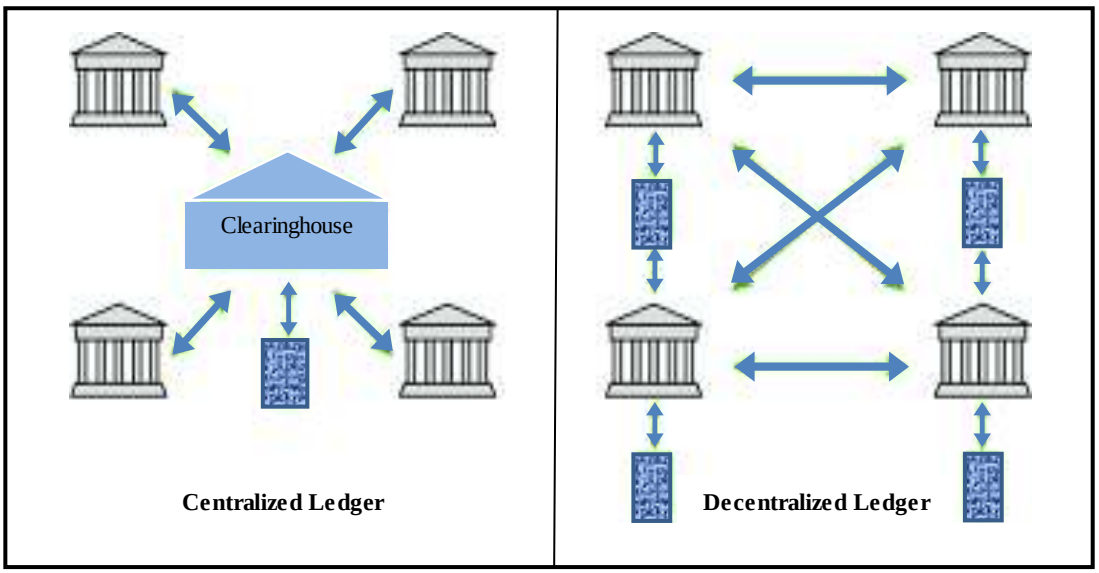


Figure (2-5). Traditional centralized ledger and a distributed ledger[41]

To solve all these problems, Distributed Ledger Technology (DLT) is used in which central authority is eliminated. DLT is commonly used in the finance industry. Sometimes, the terms blockchain and DLT are used interchangeably. Nowadays, DLT is an effective area of research in the financial field. DLTs serve as a shared database with known and verified participants. They do not have a cryptocurrency nor require mining to validate the ledger [42].

The DLT serves as a chain of blocks of data that are linked cryptographically, updated, and maintained by a decentralized network so that the data organized in a certain structure known as 'global ledger' is strong against double-spend, tampering, fraud, or other types of malicious actions. Such a strict definition, however, excludes many existing and possible future applications of distributed ledger technologies [43].

2.4.1. Types of DLT

There are three main types of distributed ledgers based on permission:

I. **Permissionless**

Anyone can validate blocks, without the need for permission from any authority. People in concern are free to maintain and work on distributed ledger systems of this type. Its systems are usually implemented as open-source software that is available for free to any person who wants to download it such as the Bitcoin application [44].

II. **Permissioned**

Participants sharing the network must be authorized and already known. This type of ledger does not need a mining or consensus mechanism to validate the ledger [42], but is gained through validation by a subset that is selected or trusted validating nodes, such as used in the proposed e-voting system[45].

III. **Hybrid**

These systems combine the benefits of a permission distributed ledger system and a permissionless distributed ledger. This gives businesses the flexibility to choose what data is needed to be public and what data should be private [44].

Hybrid distributed ledgers with the combination of private and public ledgers solve challenges on both kinds of ledgers depending on the application to be used, where private ledgers allow only the permitted users, and no one else, and Public ledgers allow any participant to view all the transactions and the data stored on the network, such as healthcare applications [46].

2.4.2. Properties of DLT

DLT is one of the most favorable inventions in information technologies with the possibility to change organizations in the economy, society, and industry fields [47].

A distributed ledger is a powerful option to be employed only in certain use cases [48]. Decentralization, Immutability, trust, security, and low cost of operations are the most general properties brought by DLTs [49], explained as follow:

I. **Decentralization:** decentralization is where no central computing devices store the ledger of transactions in one place. some network nodes will have a condition to save the ledger [18]. While protecting the anonymity of voter's identities, decentralization offers a transparent and public voting process the privacy of data transmission, and the verifiability of ballots [50].

II. **Immutability:** At the time data is written to the network, it is hard to change it back. Changing data is a challenging process and nearly impossible. This is a benefit to maintain an immutable ledger of transactions [42].

It is one of the essential characteristics that support the security and transparency of DLT and therefore guarantees its transactional integrity and suitability [51].

III. **Trusted:** Election administrators specify the election type, configure ballots, register voters, decide the lifetime of the election and assign permissions nodes, but has no control over votes nor can alter anything, the system runs in a dynamic model with no control to end until the election condition is reached [3].

IV. **Highly secured:** All transactions on the network are cryptographically secured which provides integration of data processing, uniformity, and security [52].

V. **Cost-saving:** Costs in the form of the fees which are paid to parties are eliminated [42]. As decentralization reduces the performance and cost of implementing systems on the network both in terms of operational costs as well as set up cost [25].

2.5. Cloud Services

Services used by clients and shared resources in clouds are the basis of data processing. Microsoft Azure is one of the well-known cloud computing platforms with many types of properties as shown in figure (2-6) and is considered a suitable platform for web services. Service providers can expand their services in areas where they have existing infrastructure and add new services without dealing with the main infrastructure. Cloud services are used to build, design, configure, and manage web applications and web services easily and prepare the resources needed quickly and efficiently without worrying about security detail [53].

Figure (2-6). Microsoft Azure cloud services[54].

Only an Internet connection is needed to store, post, access, or transmit information from any location, at any time in optimal response time [55].

Azure Web Sites secure platforms by encrypting the whole core of systems including data, files, databases, and servers [56]. The services also take care of load balancing and monitoring, they are containers of hosted applications and web services [57].

2.6. Hash Function

NIST "The National Institute of Standards and Technology" determines the adoption of secure hash algorithms such as SHA- 1, SHA-2 [58].

Algorithms of a hash function are used to produce the message digest during data transmission. A hash function is an essential tool for embedded security in many applications. As an input, it takes a variable-length message resulting in an output of fixed length[59].

It is hard to invert a hash value to a message input because it is a one-way function. Also, it is not possible to find a message that gives the same hash value. Any change that occurs for data in a block, for example, will lead to an invalid block because each block is related to the previous block by the proposed hash function [60].

The hash value is created using the "Secure Hash Algorithm" for example, (SHA256) generates an individual fixed-size 256-bit hash. The

SHA256 takes input for any size of plaintext and encrypts it to a 256-bit binary value, in a strictly one-way function as follow[60]:

- The in input is converted to Binary code by using ASCII and adds the value 1 at the end of the string.
- The size of the input is converted to binary code.
- The sizes of the input in binary code is added to the end of the binary input, and pad the string (adding zeros) till the whole string is 512 bit.
- The string (known as a block) is divided to 16 words (word=32 bit).
- Construct the initial hash values (H^0) that have 8 words.
- For SHA256 the initial hash values are calculated from the squared first 8 primes: Int (sqr (p) mod 1) $*16^8$), and round the results to the nearest integer.
- The result is then converted to Hexadecimal code.

The most known types of hash functions that are generally used are MD5, SHA-2. MD5 (Message Digest) is often employed in several algorithms of public key cryptographic and generally in Internet communication. It has a length of 128-bit digest for an arbitrary b-bit message [61].

The SHA-2 includes SHA-224, SHA-256, SHA-384, and SHA-512 that are suitable to be employed with the security level provided by the encryption standard [62]. SHA-256 and SHA-512 are the most used

hash functions computed with eight 32-bit and 64-bit words, respectively and have a fixed number of rounds 64 or 80, depending on which hash is used [63].

2.6.1. Properties of Hash Functions

The main properties of hash functions are as follows:

1- A Hash value is of fixed size for data of arbitrary size [64].

2- Hashing is a one-direction process. The original data cannot be obtained by moving backward [65].

3- The hash value is unique, in which the whole hash value changes if a single bit of data is changed [66]

4- Hash functions are faster than encryption, computationally [67].

2.7. Singular Value Decomposition (SVD)

The dimensionality of massive data is considered an issue that wastes space and time processing or dealing with information of any type. To solve this problem, data mining techniques or methods can play an important role in dealing with massive data, such as reducing sparse data that is less or not important, reducing the dimensionality of data without losing information that makes the process more simplified. One of the techniques that are employed in the proposed e-voting system for data and dimensionality reduction (reduces the dimensionality of a matrix without losing important data, saving time and space) is the Singular Value Decomposition technique (SVD), which is a way to analyze a matrix and decompose it resulting in a low-dimensional representation for a matrix of high-dimension. Eliminating the less

important parts makes it easier to produce an approximate representation with any desired number of dimensions [68].

SVD is well used general-purpose tool in linear algebra for data processing in data mining and machine learning fields [69]. It is used for matrix decomposition, data reduction, dimensionality reduction and it is the foundation of machine learning for feature extraction, pattern recognition, information retrieval, artificial intelligence, and other fields. Mathematically, SVD is known as the suitable low-rank approximation to a matrix of a rectangle form. The left and right singular vectors unitary matrices are mutually orthogonal, they provide the orthogonal basis for subspaces of rows and columns[70].

SVD is an efficient approach of decomposing a matrix into a collection of linearly distinct matrices, each of which has its contribution to energy [71]. Any 2-dimensional matrix A of size ($m \times n$) can be factorized into three matrices. Where m represents the rows, n represents the columns, and $m \geq n$ [72] [73]. The result of multiplying the three matrices is approximately equal to the original matrix A, as explained in the equation (2.1), (2.2), and (2.3) [73] [74].

$$\mathbf{A = USV^T} \qquad (2.1)$$

$$= \begin{bmatrix} u_{1,1} & \cdots & u_{1,n} \\ u_{2,1} & \cdots & u_{2,n} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & \cdots & \vdots \\ \vdots & & \vdots \\ u_{n,1} & & u_{n,n} \end{bmatrix} \begin{bmatrix} \sigma_{1,1} & 0 & \cdots & 0 \\ 0 & \sigma_{2,2} & \cdots & 0 \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & \cdots & \vdots \\ \vdots & & & \vdots \\ 0 & & & \sigma_{n,m} \end{bmatrix} \begin{bmatrix} v_{1,1} & \cdots & v_{1,m} \\ v_{2,1} & \cdots & v_{2,m} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & \cdots & \vdots \\ \vdots & & \vdots \\ v_{m,1} & & v_{m,m} \end{bmatrix}^T \quad (2.2)$$

$$= \sum_{i=1}^{n} \sigma_i \, u_i \, v_i^T \quad (2.3)$$

U and V are vector matrices where $U^TU= UU^T= I$ and $V^TV=VV^T=I$ and S is a diagonal matrix where only the diagonal singular values are non-zero values in which: $\sigma_1 \geq \sigma_2 \geq \sigma_{3.....} \geq \sigma_m \geq 0$

When $n \geq m$, the matrix S has at most m non-zero elements on the diagonal. Therefore, it is possible to exactly represent the original matrix A using the economy SVD as shown in equations (2.4) and (2.5) [75].

$\sigma_1 u_1 v_1 + \sigma_2 u_2 v_2 + \ldots\ldots\ldots + \sigma_m u_m v_m$ where $r = m$ (2.4)

$$\mathbf{A= \hat{U}\hat{S}V^T} \quad (2.5)$$

If we have very low σ, then the low singular values σ can be truncated and the truncated SVD may still be exact as shown in equation (2.6) and (2.7), Which is proved mathematically by Ekard-Young Theorm (1936) [75]. Where:

$\sigma_1 u_1 v_1 + \sigma_2 u_2 v_2 + \ldots\ldots\ldots\ldots\ldots + \sigma_k u_k v_k$ where $r = k$ (2.6)

$$\mathbf{A = \tilde{U}\check{S}\tilde{V}^T} \quad (2.7)$$

With rank= $k \leq m$, the left singular vectors (Matrix U) is $n*k$, the right singular vectors (Matrix V) is $k*m$, and the matrix of singular values is the sub-block of $k*k$.

In the truncated SVD the property of $\tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{U}} = \mathbf{I}_{\mathbf{k*k}}$, but $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^{\mathbf{T}} \neq \mathbf{I}_{\mathbf{k*k}}$ because the identity matrix is of size $n*n$ which is not true for $I_{k*k.}$

By that, the huge matrix can be represented as a multiplication of three small matrices after truncating SVD, which can represent the exact matrix, used as the best low-rank approximation such as in image compression. The truncated SVD is shown in figure (2-7).

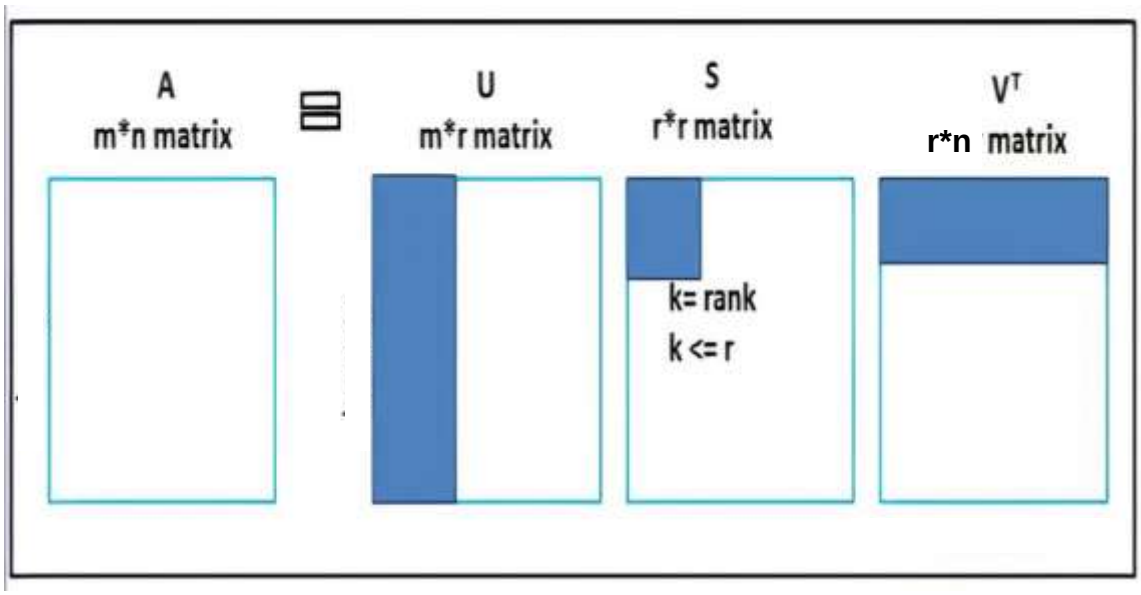


Figure (2- 7). Size reduction of SVD decomposition [71].

SVD serves as a powerful tool in many applications, it can be used to achieve a smaller k which is the suitable rank [76]. When the goal is *exact* Binary Matrix Decomposition (BMD), by $A \in \{0, 1\}^{m \times n}$, where A is an $m \times n$ Binary matrix. BMD tends to find two matrices $U \in \{0, 1\}^{m \times k}$ and $V \in \{0, 1\}^{k \times n}$ such that the difference $\{M - U \circ V\}_L$ under

some norm L is minimized with a given k or smaller as possible. That small k is the *Boolean rank* of a binary matrix A that may be smaller or larger than its real rank [77].

The *exact* BMD is satisfying and pleasing that it is useful for many applications in the future. However, it can be used for approximate BMD, by finding the product of $U \circ V$ that includes all the 1's and no 0's in A. This is often known as the "from-below" approximation [78].

2.7.1. Some Applications on SVD with Matrices

I. Low-Rank Approximation of Matrices

Many fields of science such as effective rank, data compression, and image processing, Also engineering, approximating a matrix by a lower rank matrix according to a given norm is desired. The truncated SVD is simply an easy way for this matter [75][79].

II. Using Singular Value Decomposition in Image Compression

In general, the Singular Value Decomposition (SVD) decomposes the original matrix into three matrices. The aim is to approximate the data-set of high dimensions using fewer dimensions. SVD displays the substructure of the high dimensionality original data by reducing it into data of lower-dimensional and arranges the data from most to the least variation[80].

SVD factorizes the $m*n$ original matrix (m rows and n columns) into three matrices, which can be written as $A = U\Sigma V^T$ where U_{m*m} and V^T_{n*n} are orthogonal matrices known as left singular matrix and right singular matrix of A respectively and Σ is a diagonal matrix of non-

negative real numbers known as singular values of *A* in the order $m \times n$ [81] [82].

SVD of a given matrix is implemented on a matrix *A* as follows:

- AA^T and A^TA are calculated.
- AA^T is used to form U, which is found by calculating eigenvalues and eigenvectors of AA^T.
- V is found as U by calculating the eigenvalues and eigenvectors of A^TA.
- The columns of U and V are formed by dividing each eigenvector by its magnitude.
- Singular values are then computed by founding the square root of eigenvalues. They form a diagonal matrix arranged in descending order.

III. Determination of the Effective Rank

The SVD is also used to specify the actual rank of a matrix. This is performed by counting the number of singular values that are above a certain tolerance, τ. The tolerance $\tau = 0$ is used for the actual rank and some small number determined by the user according to the application that is used for the numerical rank (i.e., $\tau > 0$ for numerical rank) (e.g., $\tau = \varepsilon \, ||A||_2 = \varepsilon\sigma_1$ where ε is machine precision). The actual rank of a matrix is defined as the number of singular values $\varepsilon > \tau$, $r(A)_\tau = \{ k: \sigma_k (A) > \tau \, , \sigma_{k+1} (A) \leq \tau \}$ [79].

Chapter Three

The Proposed System

3.1. Introduction

This chapter describes the e-voting system by employing an adaptive dimensionality reduction technique known as singular value decomposition (SVD) to produce a ledger.

The proposed system covers important features, it is a transparent web application system that allows every citizen to view the results in real-time without the lack of delay in days till announcing the results, less cost is needed for materials or finance issues. The system is trusted as there is no control over the system as it starts dynamically in real-time. Anonymity is an important feature for all citizens to vote safely without exposing their identity during the voting process. Also, security details are covered in several stages within the back-end platform and other cloud security services. However, our system uses the hash function SHA256 to protect votes within a block structure to avoid any change that may occur by the administrator.

The rate of transparency (viewing the results in real time) and the verifiability of the overall system is based on the SVD technique that is employed as a matching tool, where a copy of the results is transformed to another form of three matrices distributed in separate places that are retrieved back to the original result matrix and matched with the SQL DB results. This process plays an important role in evaluating the rate of success of the election event.

3.2. Settings of the Proposed System

Before explaining the general design of the proposed e-voting system, an explanation of the needed settings that are used in the propsed system are as follow:

3.2.1. Microsoft SQL Software Management

Microsoft SQL software management is used to manage the databases of the voters and candidates. It is secured and connected to the Azure cloud service. Processes applied to any database should be within the IP range determined in the cloud service that secures the database data and files.

3.2.2. BLOB Storage

Our system takes advantage of cloud services as an environment to save the distributed ledger as a blob. A blob refers to binary large objects. Blobs are files that store big data. Blobs are viewed as folders, they are stored in containers that group a set of blobs. A storage account can contain any number of containers, and a container can store any number of blobs [74]. Because of its high availability, flexibility, security, and low cost we used blobs to store the outputs of SVD (U left matrix, S singular value matrix, and V right matrix) that are distributed in different servers as blobs.

3.2.3. Queue Storage

A queue is used to gather transactions that arrive at the same time, and to avoid any shutdown problems due to a collision that may happen

to the system by casting votes all at the same time, the queue will have a short time delay to manage the arriving transactions.

3.2.4. Cloud Servers

The cloud servers are used to store and manage the files and data in different places around the world to avoid locality. In which a group of resources is used on our system as follows:

- The application service is stored in a server in South Central United States.
- The SQL server and database in the North of United Arab Emirates.
- The Queue storage account is stored in a server in the East of the United States.
- The Blob storage account for the U matrix of SVD is stored in a server in East Asia.
- The Blob storage account for the S matrix of SVD is stored in a server in North Switzerland.
- The Blob storage account for the V matrix of SVD is stored in a server in Central Canada.

3.3. The Working Mechanism of the Proposed E-voting System

The e-voting system works as follows:

1- The e-voting system is a web application that runs dynamically in real-time based on a specific time, set by the administrator.

2- As soon as the system begins to run, a connection to the cloud services is opened after passing a firewall service based on the acceptable IP address for accessing databases, storing results, and receiving/casting information from/to the distributed network nodes that hold the results for each block.

3- The administrator has no control over the system, in which the system can not be stopped temporarily or permanently for any reason till the end of time for the election event.

4- The votes are gathered as transactions in the form of blocks that are secured by the hash function SHA256, and all blocks are chained based on a previous hash value to secure votes.

5- The SVD technique is used for the first time in an e-voting system as a distributing, matching, and dimensionality reduction tool to serve as a suitable structure for DLT.

6- The system is also secured by cryptography methods for all levels by employing the AES256 algorithm as a service that encrypts data, files, storage accounts, and all related resources for the e-voting system.

7- The system stores an immediate result of votes for each block and applies SVD immediately to distribute the results in another form of data and cast them in a distributed manner to separated network nodes in different servers.

8- Based on an adaptive SVD that is applied to construct an incremental ledger for each block, the final ledger is used to count the votes of each candidate and match the votes with the SQL database of results.

9- The election results are distributed and not locally saved. In other words, data and important details are stored as another copy and secured using cloud services.

3.4. The General Design of the Proposed E-voting System

The system consists of four main phases, the preprocessing phase, confirmation phase, e-voting phase, and the result phase, some phases have stages with many steps as shown in figure (3-1).

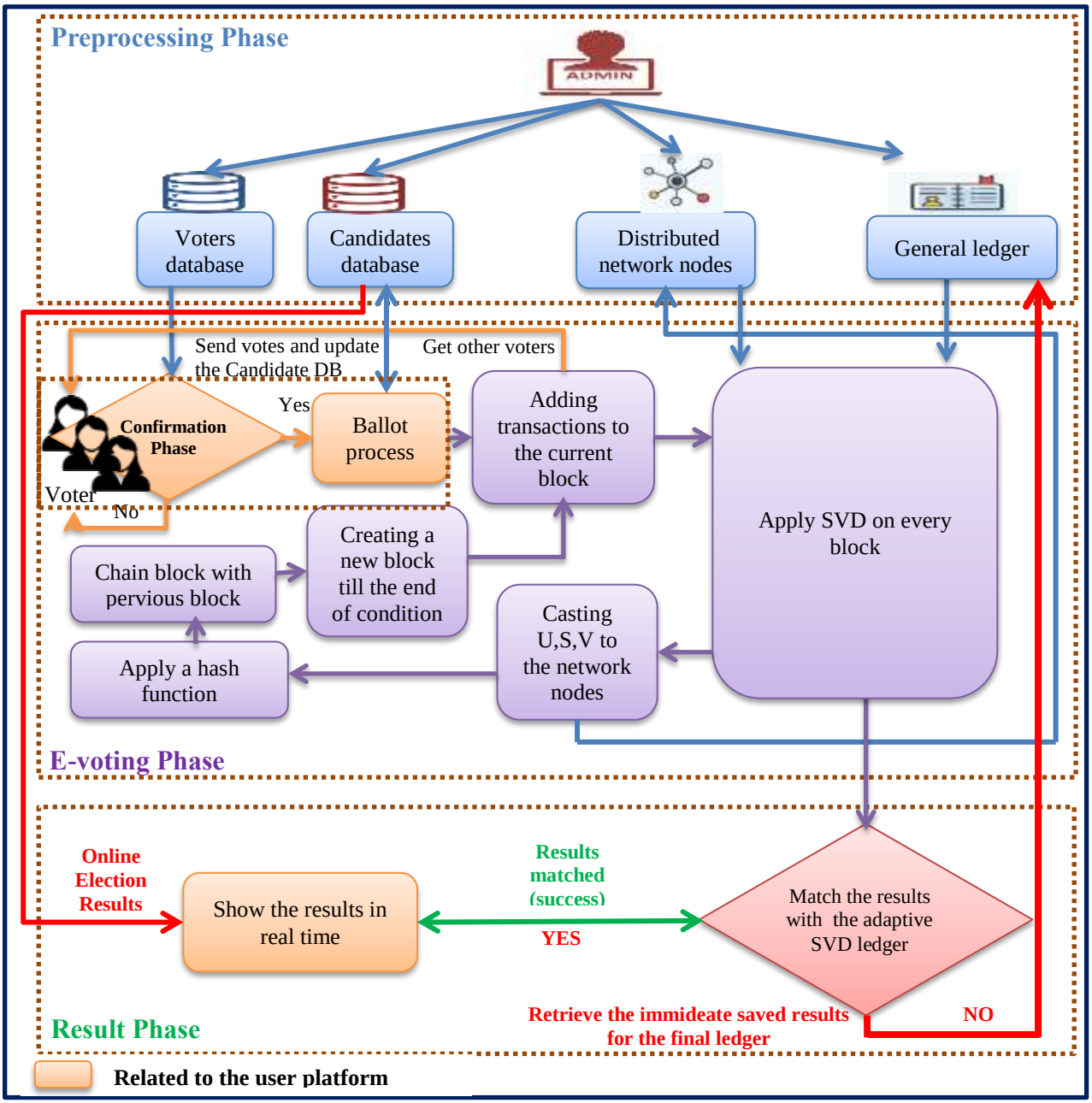


Figure (3-1). The block diagram of the e-voting system.

The proposed system is built for concerned participates only, data is time-stamped and protected by a hash function, providing a degree of decentralization. The system generally is divided into two platforms, the user platform, and the administrator platform, each platform has its share of phases. The phases and stages are explained within the platforms as follow:

3.4.1. The User Platform

The user platform is concerned with all citizens who have the right to vote. As soon as the administrator releases the URL of the web application to citizens from the cloud application service, the confirmation and voting process begins. The user platform allows citizens to vote till the event reaches the time set by the administrator referring to the end of the election event. The general design of the user platform is shown in figure (3-2).

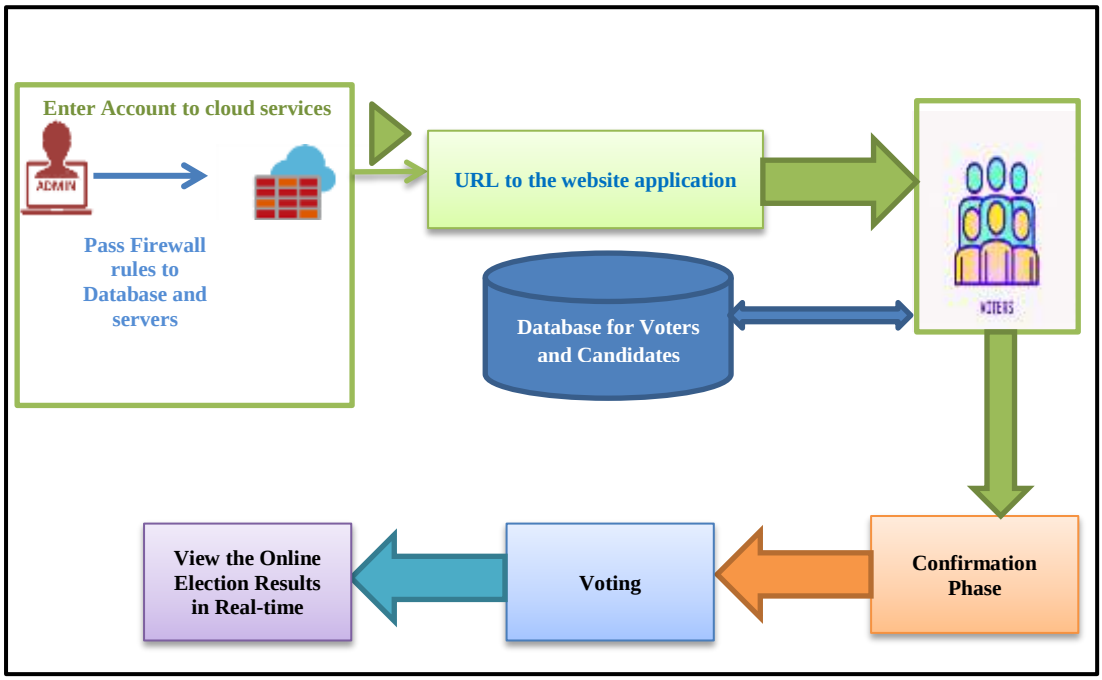


Figure (3-2). The general form of the user platform

The user platform consists of the following phases and stages:

I.Preprocessing Phase

In this phase, the list of candidates and voters database is prepared for users to enter the e-voting website and vote for the desired candidate. The candidate database includes the candidates' names, IDs, counts, and photos. While the voter database includes the citizens' name, ID, and state which is indicated by 1 for a vote or 0 otherwise.

II. Building the Distributed Network Nodes

This stage includes setting the distributed network nodes in three separate servers around the world in which they are responsible for storing, receiving, and retrieving the outputs of the SVD ledger (matrices U, S, and V^T) where the votes are stored and processed as another form.

III. Generating The General Ledger

The general ledger initially contains zero values as a matrix representing voters and candidates. In other words, the ledger represents the number of votes for each candidate, where the candidates are represented in rows and each column represents a citizen that is allowed to vote only once.

Each column should contain many zeros and at most only a single 1 representing a vote. When a block containing votes is generated, counts are added to the SQL database (The online election results in real time) for candidates in this block and the general ledger is updated. The ledger is processed by the SVD technique, it is applied to transform the results into another form and distribute them to the network nodes. The

initial general ledger and distributed network nodes are illustrated in figure (3-3).

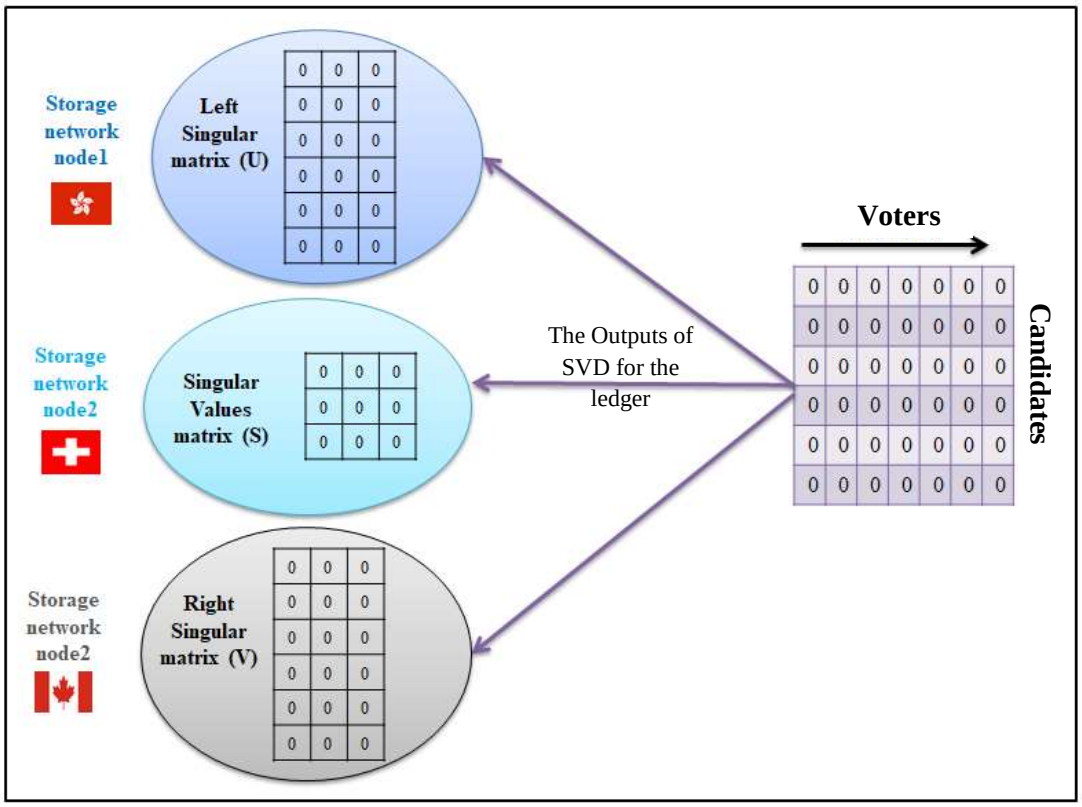


Figure (3-3). An illustration of the initial ledger and the distributed network nodes.

IV. The Confirmation Phase

This phase is where each citizen is allowed to vote based on the international ID card and city then cast a vote for only one time. The confirmation process includes checking the ID of the voter, ensuring that the voter votes only once and cannot change his/her vote based on a valid ID. The confirmation process block diagram and algorithm are shown in algorithm (3-1) respectively.

Algorithm name: Confirmation process

Input: Citizens' ID and city.
Output: Reject the input or show the results in real time.
Begin
1. **If** the ID of the voter does not exist in SQL database **then**
2. ID is rejected //message appears asking for a valid ID
3. **Else**
4. **If** the voter has previously voted **then**
5. The voter can go to the result page
6. **Else**
7. Voting for a candidate
8. Assigning a flag as voted /*Change the state of the voter indicating that the voter has voted */
9. Increasing the counter of the selected candidate by one.
10. **End if**
11. **End if**
12. The voter can go to the result page.
End

Algorithm (3-1). Confirmation process algorithm.

V. The E-voting Phase

The e-voting phase related to the voter in the user platform is the ballot process stage which is a web page interface for the list of candidates that appears to the voter after passing the confirmation process to vote for the desired candidate.

The votes are stored temporarily in queue storage, which aims to hold the votes according to their precedence. There is a short time delay between votes to manage the flow of votes in the queue, in case a

massive number of votes arrive at the same time. The illustration of votes arriving at queue storage is shown in figure (3-4).

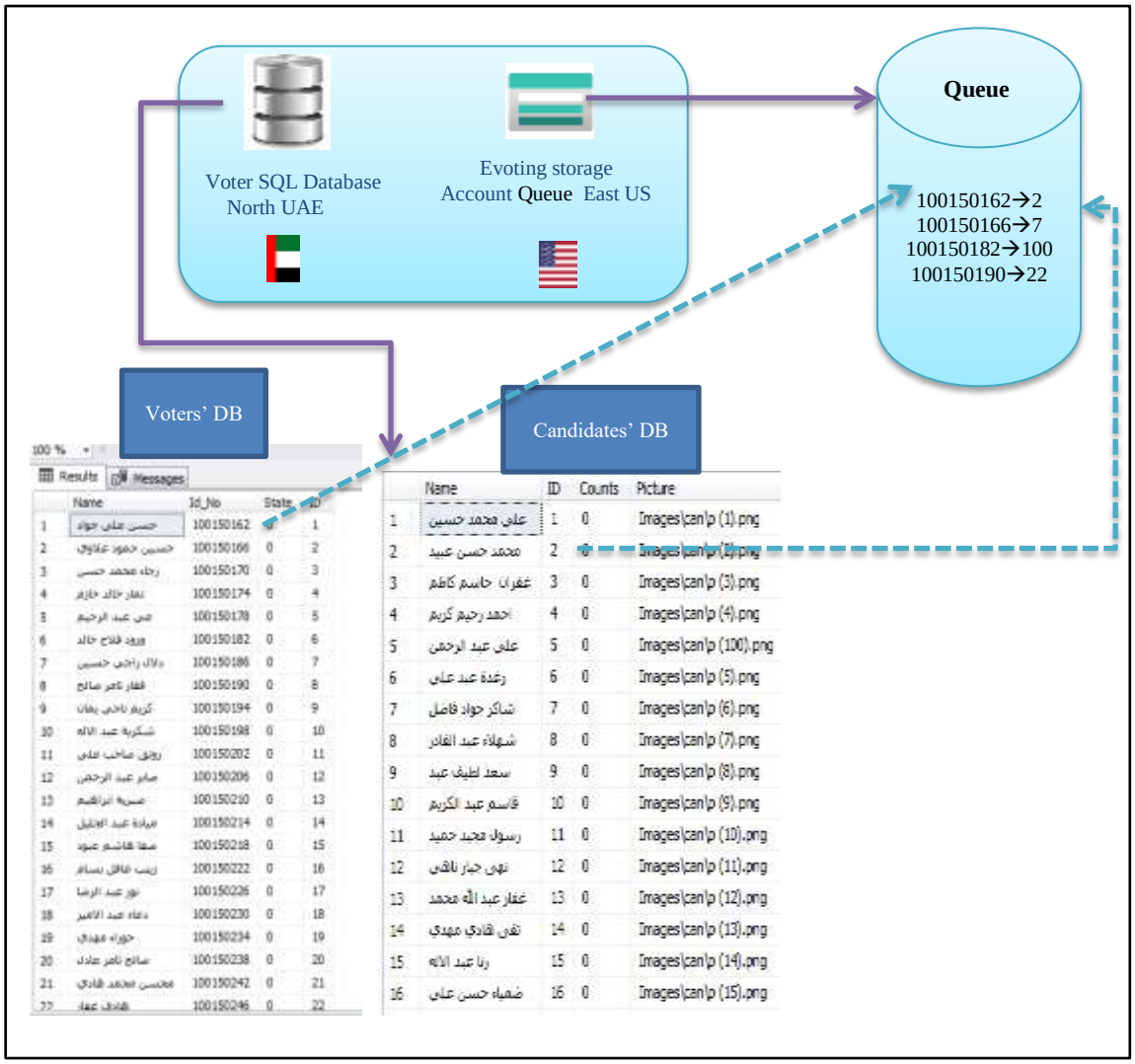


Figure (3-4). The queue storage for the e-voting system.

The votes are kept as transactions in the queue storage till the administrator opens a connection to the cloud storage and keeps listening to check the queue. Whenever there are transactions, they are sent to the administrator platform that receives the votes based on the number of transactions determined by the administrator as soon as the

system begins to run. The transactions are then removed from the queue and not allowed to be repeated.

IV. The Result Phase

The final phase in concern to the user is the result phase that can be a chart or a table suitable to show the results online and viewed by all citizens showing the counts for each candidate to achieve transparency. All the votes are stored in a queue cloud storage and sent to the administrator platform to be processed by applying the SVD technique as a data dimensionality tool and as a matching tool that saves the results immediately in another form composing a ledger that is saved and distributed to cloud storage in separated servers.

3.4.2. The Administrator Platform

The administrator platform runs in real-time mode, in which all processes execute dynamically within time without any control over the system. The only step needed to be made by the administrator to start running the system is to enter the desired number of transactions for a block. Figure (3-5) illustrates the administrator's interface and algorithm (3-2) states the steps performed in the administrator platform.

The stages in concern to the administrator are explained as follow:

I. Connecting the Cloud Resources

Before running the system, a connection to the cloud services should be opened to reach the resources that are used in our system.

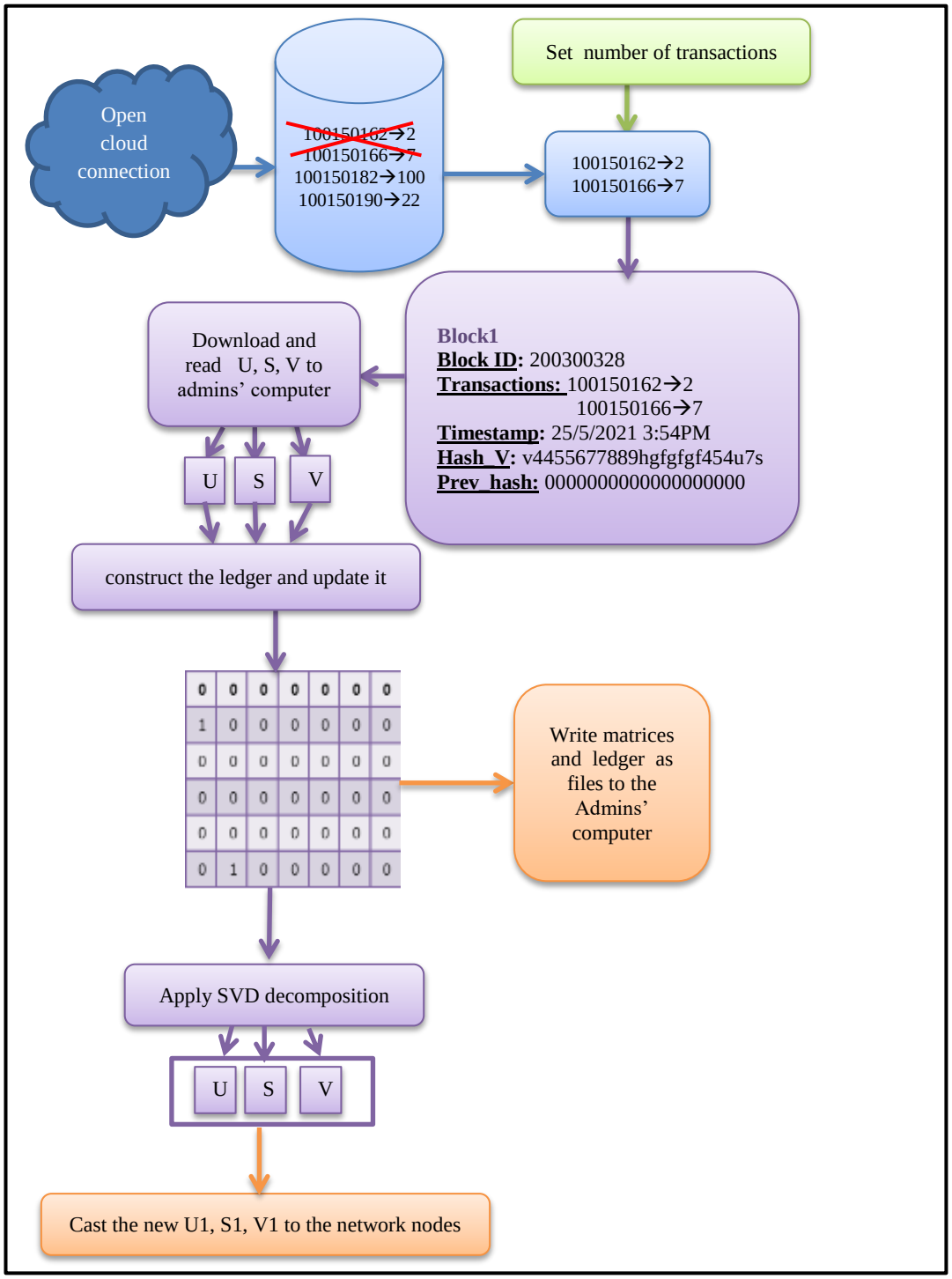


Figure (3-5). The Administrator platform with two transactions.

Algorithm name: Administrator platform

Input: Number of transactions, A vote, the initial distributed network nodes, and the initial general ledger
Output: The matrices, U,S, and V, A Block of transactions (incremental updated ledger)
Begin
1. Open connection to the cloud resourses
2. A Counter to count the blocks
3. Enter the number of transactions (votes) needed in a block
4. **For** i=1 to number of transactions
5. **Begin**
6. Assign the block an ID. /*The id of a block is the summation of the voters' ids added to the block */
7. Add votes as transactions to the current block.
8. Add a time-stamp for the block.
9. Apply SHA256 to the block.
10. Link the block to the previous block.
11. **End for**
12. Retrieve U, S, and V from the distributed network nodes
13. Extract the ledger // multiply U*S*V'
14. Validate the transaction /*check that the voter voted only once*/
15. Update the ledger
16. Apply SVD decomposition
17. Cast the new U, S, and V to the distributed nodes
18. Save a copy of the current SVD ledger for the current block
19. Increase the counter for Blocks by 1
20. **Repeat** steps from 2 to 19 for the next block till the end of time
End

Algorithm (3-2). Administrator platform algorithm.

To deal with data in the databases within the system, such as reading, updating, or fetching data, we need to open a connection string to the cloud storage to reach the stored data.

Databases can be managed from Microsoft SQL server management by passing through the firewall rule which is the range of IP or IPs allowed to start a connection, If the IP is not with the range determined by the admin a connection failure occurs as shown in figure (3-6).

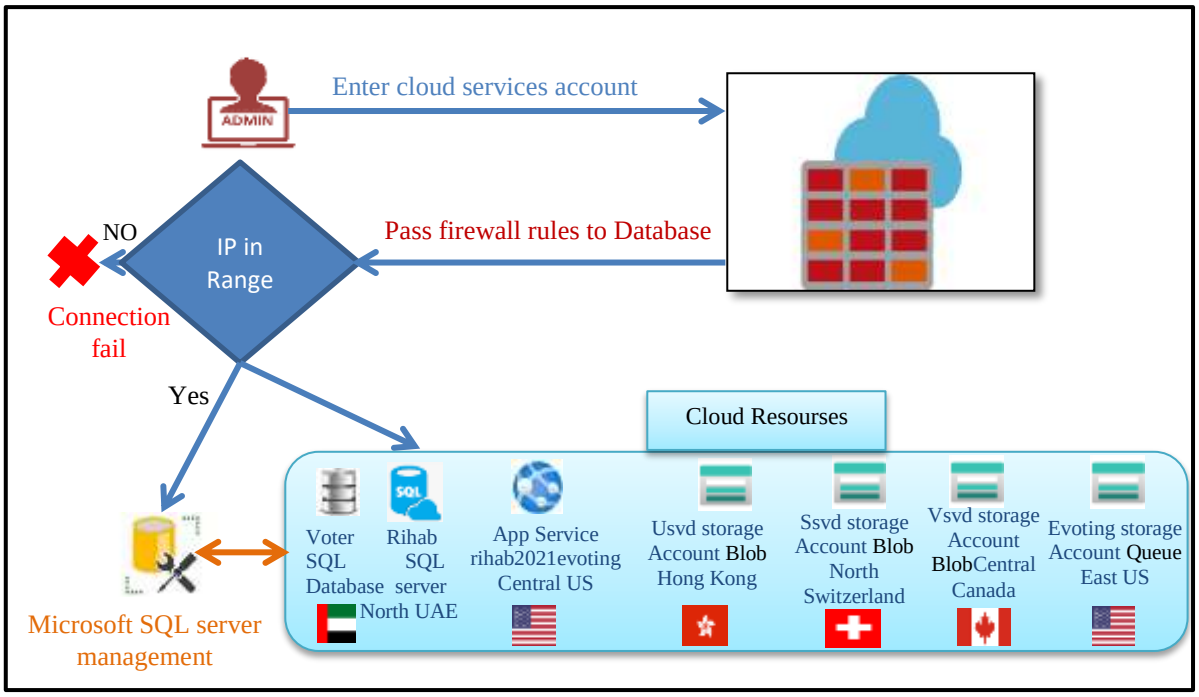


Figure (3-6). Connecting the cloud services.

The type of storage used to stores the votes is queue storage. Whenever a vote occurs, the voter's ID is fetched from the voter database and a count is added to the chosen candidate, this information is added in the queue storage.

The queue storage is considered a reference to the admin to check which ID voted to which candidate number. Composing a database that is based on the queue contents for the surveillance system of the administrator.

Other storage services are blobs, which are files for binary large objects that are used to store the contents of the SVD outputs, U for the left singular matrix, S for the singular matrix, and V for the right singular matrix. The BLOB files are sent and received between the system and cloud storage during the execution of the event and saved in separate servers.

II. Determining the Number of Transactions

After running the system, the administrator determines how many transactions are to be added in a block. Each vote represents a transaction that is fetched from the queue storage service and the transactions that are fetched from the queue are equal to the number of transactions set by the administrator for each block.

Generally, there can be two ways to determine the number of transactions in a block, explained as follow:

1. The administrator determines manually the number of transactions in a block, by entering the number of transactions after starting to run the system.
2. The transactions are added to a block automatically after a fixed period. For example, after 1minute add all transactions in a block and so on.

Adding transactions manually is more efficient than adding them automatically since some blocks may contain no transactions, which is created as a useless block, some blocks may contain few or a lot of transactions. For this reason, our developed e-voting system prefers adding transactions manually in advance, so that all blocks have the same share of transactions with no waste of blocks with empty transactions.

III. Surveillance Stage

After starting the connection to cloud resources, the administrator platform behaves as a surveillance system in which the system listens to the queue storage that contains transactions of votes, one after the other determining which ID voted to which candidate. The system keeps fetching transactions (votes) based on the number of transactions determined by the admin in advance to create a block.

IV. Block Creation

When the number of transactions is enough to create a block, the block is assigned an ID which is the summation of all voters' IDs in that block, and contain transactions (votes) indicating the IDs for each voter that voted for candidates by their IDs, a time-stamp indicating the date and time for creating the block, a hash value, and the hash value of the previous block. The hash function as explained in section (2.6), is used to chain the blocks together, making it more secured and more complicated against any attempt of manipulation. In which, each block

is chained by the previous block by its hash value adding more security to the blocks.

V. SVD Stages

After the creation of a block, a connection string to the BLOB storage files is opened to retrieve the three matrices U, S, and V that are used to store the results in another form processed by the SVD technique. Initially, the three matrices U, S, and V^T (outputs of SVD) and the general ledger are of zero values. The three matrices are based on the general ledger that is in turn based on the blocks' information (transactions of votes).

By multiplying U, S, and V^T, the SVD ledger is constructed according to the information of transaction in the block, update the matrices, and cast them back to be stored as blob files in separated servers. Figure (3-7) and (3-8) shows the general use of SVD in the proposed e-voting system, the steps of the SVD stage for a single block.

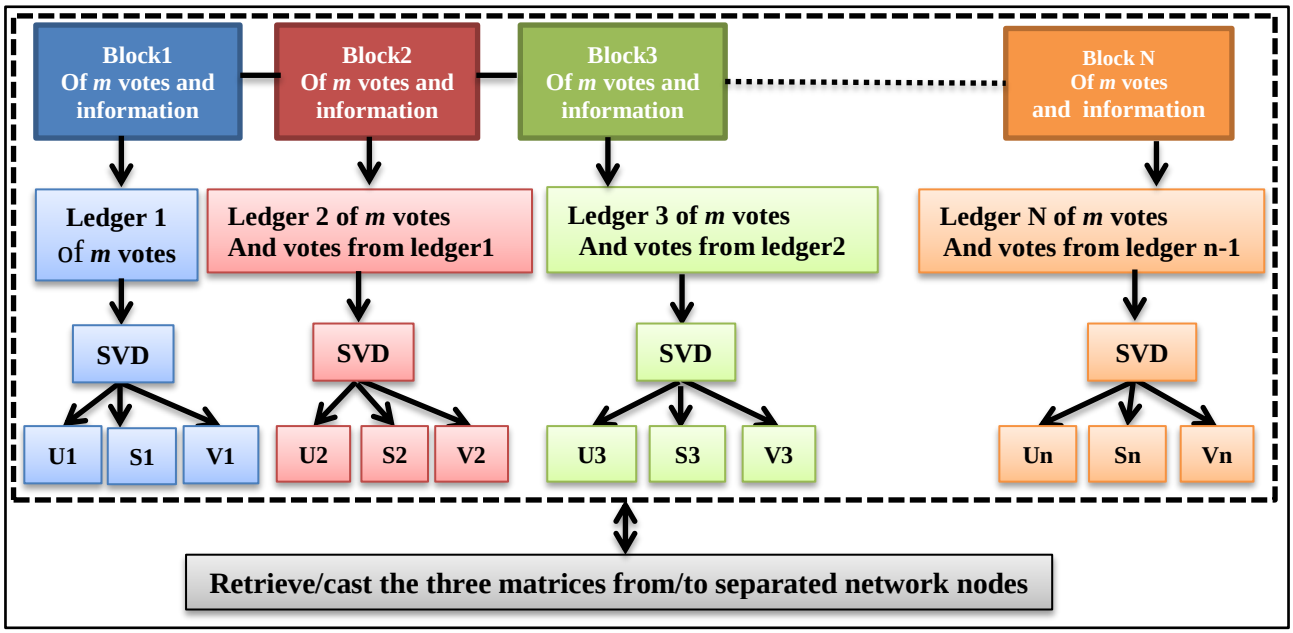


Figure (3-7).The general use of SVD in the proposed e-voting system

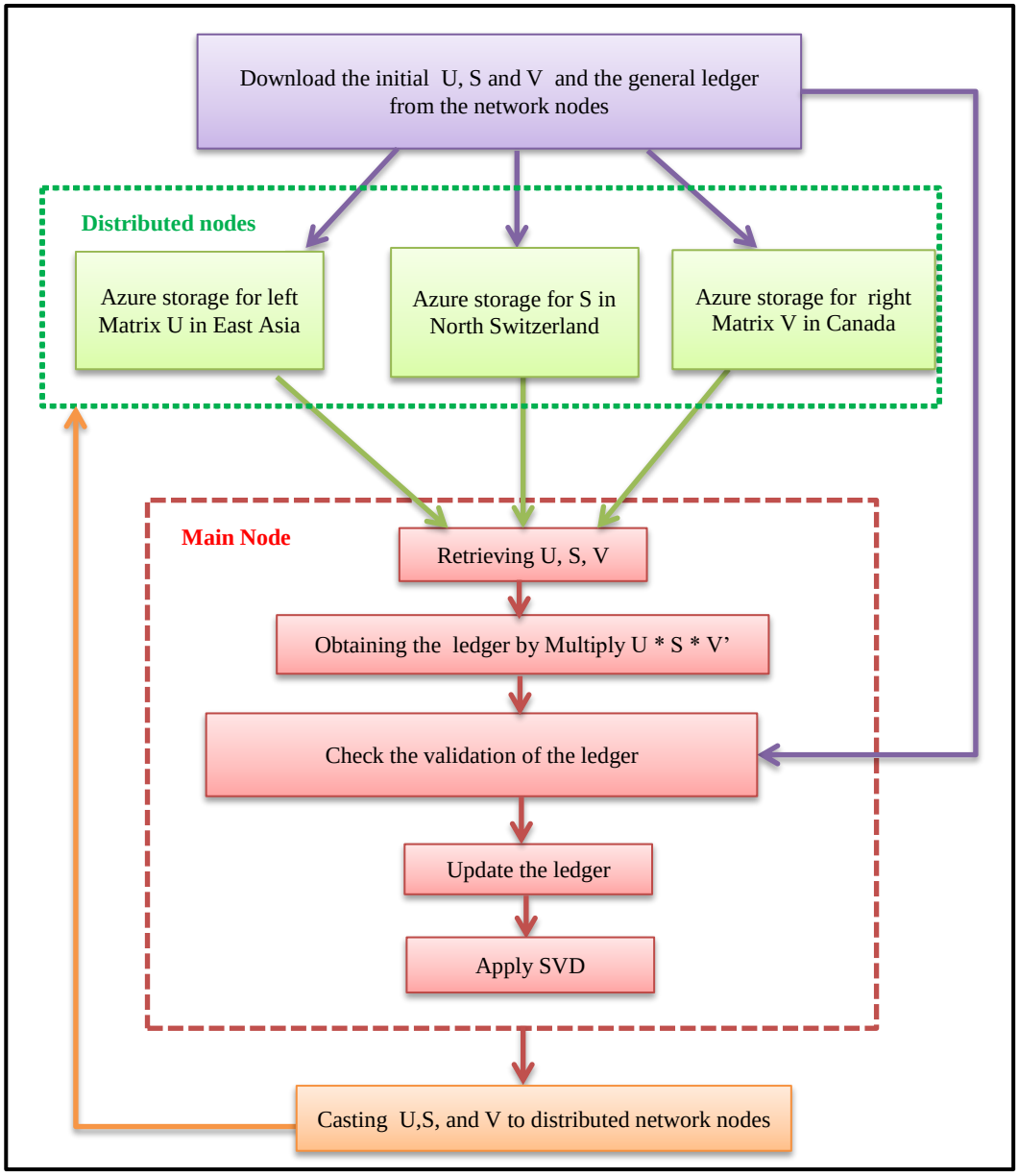


Figure (3-8). A Block diagram showing the SVD stage for a single block.

The steps performed by the SVD stage are explained as follow:

1. Retrieving U, S, V from the network nodes

The system employed three distributed network nodes used as the cloud storage nodes (one for the left matrix of SVD decomposition (U), one for representing the singular values of SVD decomposition (S) and

the third one represents the right matrix of SVD decomposition (V^T)). The fourth node is the main node for processing, retrieving, and casting the information after creating each block. This process allows us to store the results in three separated matrices due to the SVD technique that distributes its outputs to three matrices instead of having one massive ledger of results which saves us time and space in retrieving and casting operations.

2. Obtaining the Ledger

The ledger is obtained through the individual matrices that are stored in different servers as nodes, U, S, and V, by multiplying U, S, V^T of the current block in the main node.

3. Ledger Validation

In this step, the ledger is checked to be validated. In other words, each row that represents a citizen in the general ledger must have at least only one 1 corresponding to only one candidate.

4. Updating the General Ledger

After checking the validation of the ledger, the ledger for the current block is updated in the main node to hold the current data in an acceptable form.

5. Applying the SVD Decomposition

The SVD method is applied to produce three coefficient matrices U, S, and V in the main node to deal with a ledger of smaller size instead of the whole ledger with sparse data, saving time and space.

VI. Casting U, S, V to the Network Nodes

After applying the SVD method on the updated ledger, the main node cast back the individual matrices U, S, and V to the distributed network nodes.

VII. Creating a New Block as the Current Block

The new block is created as the first one, by adding votes as transactions with other information related to the block, and again the ledger for the new block is managed by the distributed network nodes and processed by the main node to validate, update and cast U, S, and V of the ledger.

All these steps in the Administrator platform keep running until the election ends. The SVD dimensionality reduction technique is applied to the ledger for each block and the reason for all these processes is to have another copy of the results that are saved as SQL database in another form which is achieved by applying the SVD technique and distribute the outputs of SVD matrices as storage nodes in the form of blob files.

The ledger is always distributed to the separated servers and stored incrementally after the creation of each block. In other words, there exists a ledger that is incremented and updated including the new and previous information of the previous ledger every time a block has been created as shown in figure (3-9).

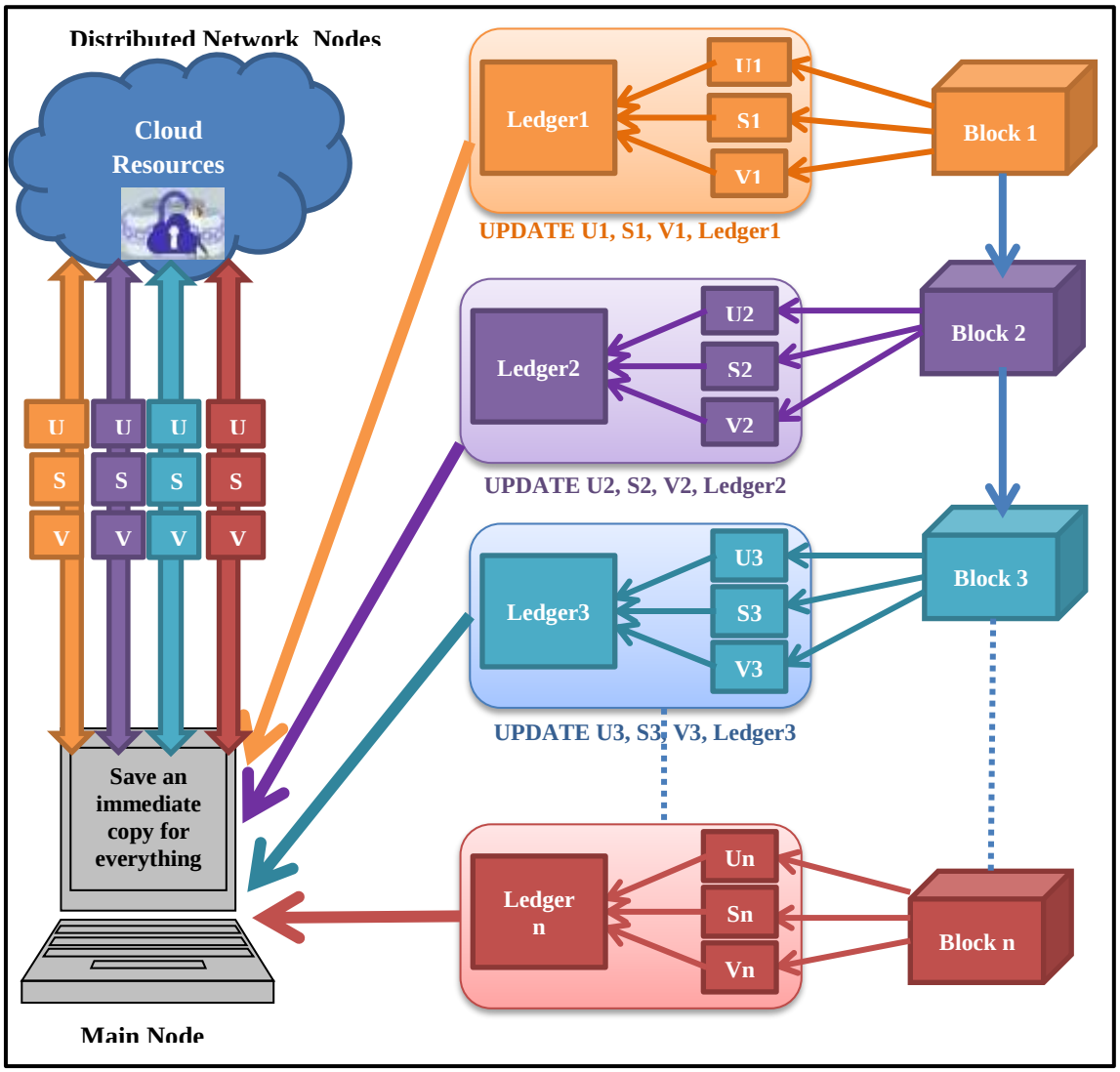


Figure (3-9). An illustration of the distributed incremental SVD ledger

At the end of the election event, the final incremented SVD ledger is used as a tool that is matched with the election results. This step is important in case the SQL database of results (election results) is manipulated, where the SVD ledger is a copy of the original results in another form stored in a distributed manner.

Every time a ledger is constructed, it is an updated ledger containing the information of the previous ledger and so on till the final ledger is constructed. Figure (3-10) shows the incremented ledgers with their components and the final ledger by matching the counts with the online election results (SQL database).

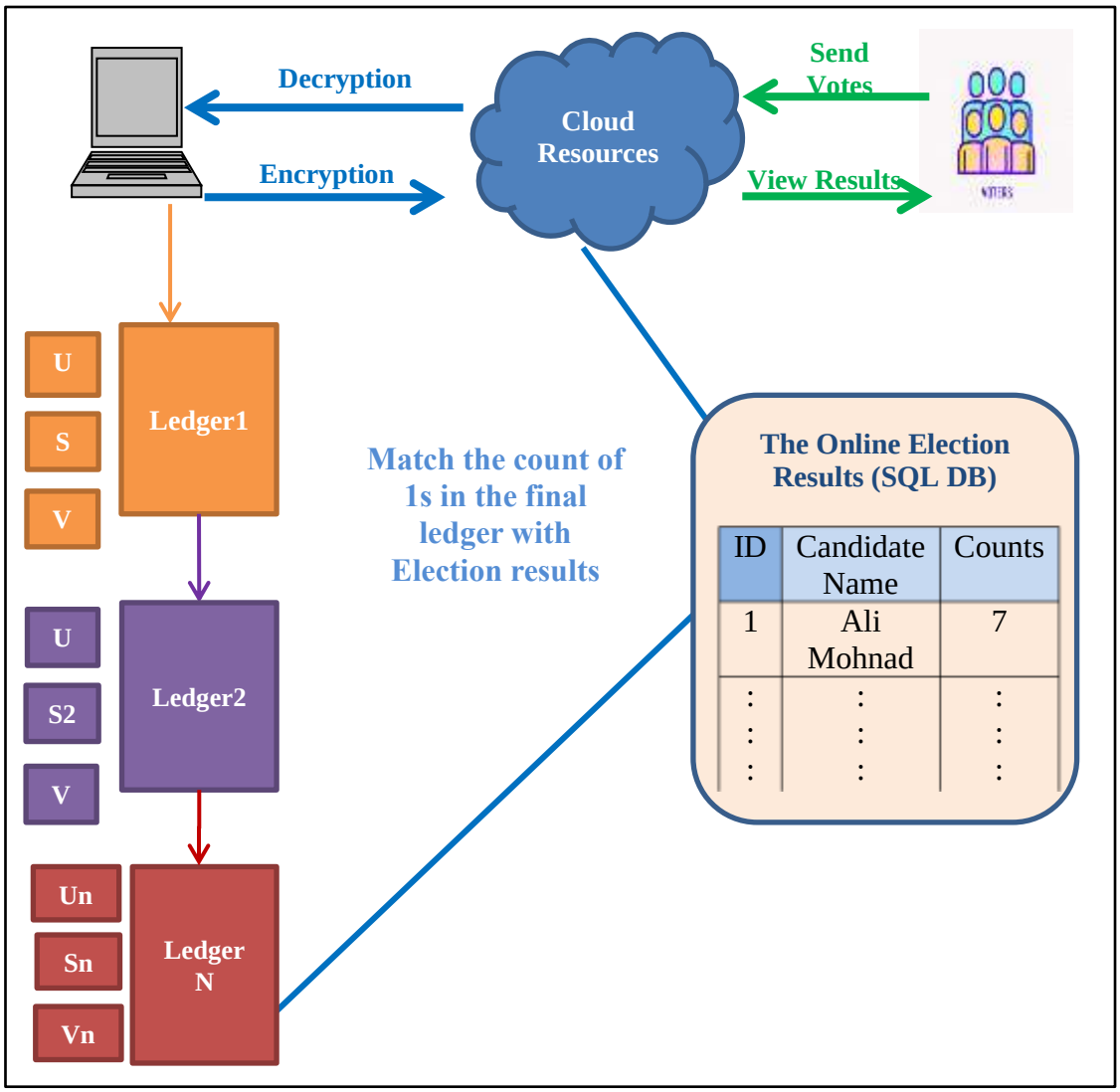


Figure (3-10). The results phase for matching the results.

The developed e-voting system works in hard real-time where all processes should run efficiently in a dynamic manner. The system starts

to run on the day of the election event, in which all tasks of the administrator platform execute within real-time, meaning that the system is time-dependent. also, all web pages change dynamically according to the flow of the event, such as the main and result web pages that change their information every time a voter enters the system.

The surveillance interface of the administrator keeps listening to the fluence of the system till the time determined within the system is terminated, even the administrator has no control over the system.

The system keeps running, adding and creating blocks, updating the SQL database showing the results directly and creating a copy of the results based on the SVD technique which transforms the results in another form, distributing the SVD ledger we created and employ it as a matching tool against the SQL database to match and evaluate the rate of transparency and success for the election event in Iraq.

A dynamic system should deal efficiently with all the components of the system including space and time, committed to a fixed rule that describes the time dependence set for the event.

Appendix (A) shows a full guide for the system interfaces and an actual example made by volunteers to illustrate how the system works from the view of the user and the administrator platform.

Chapter Four

Experimental Results

4.1. Introduction

This chapter displays the topology of the proposed e-voting system. Also, it shows the

experimental results for SVD employed with Binary matrices in a developed e-voting system with an incremental ledger, and in general cases where the matrix or ledger is altered all at once.

The proposed system solves the issue of dealing with a ledger of the election results by applying the SVD dimensionality reduction technique that is employed also as mentioned previously as a distributing and matching tool.

The experimental results are implemented on a Lenovo machine, Intel Corei5, CPU 2.5GH with 8GB of RAM, using visual basic, C#, and ASP.NET through visual studio2019.

4.2. The Topology of the E-voting System

The practical example that illustrates the work of the e-voting system includes several stages. Starting from the cloud services of type PaaS (Platform as a Service), where storage accounts and hardware resources are managed. Figure (4-1) shows the topology of the whole system. The stages of work are as follows:

1- A connection to the cloud account should be opened bypassing the server and data firewall rules. These rules are set by the cloud services automatically, such as checking the IP that should be within the range of IPs allowed to enter the desired resources.

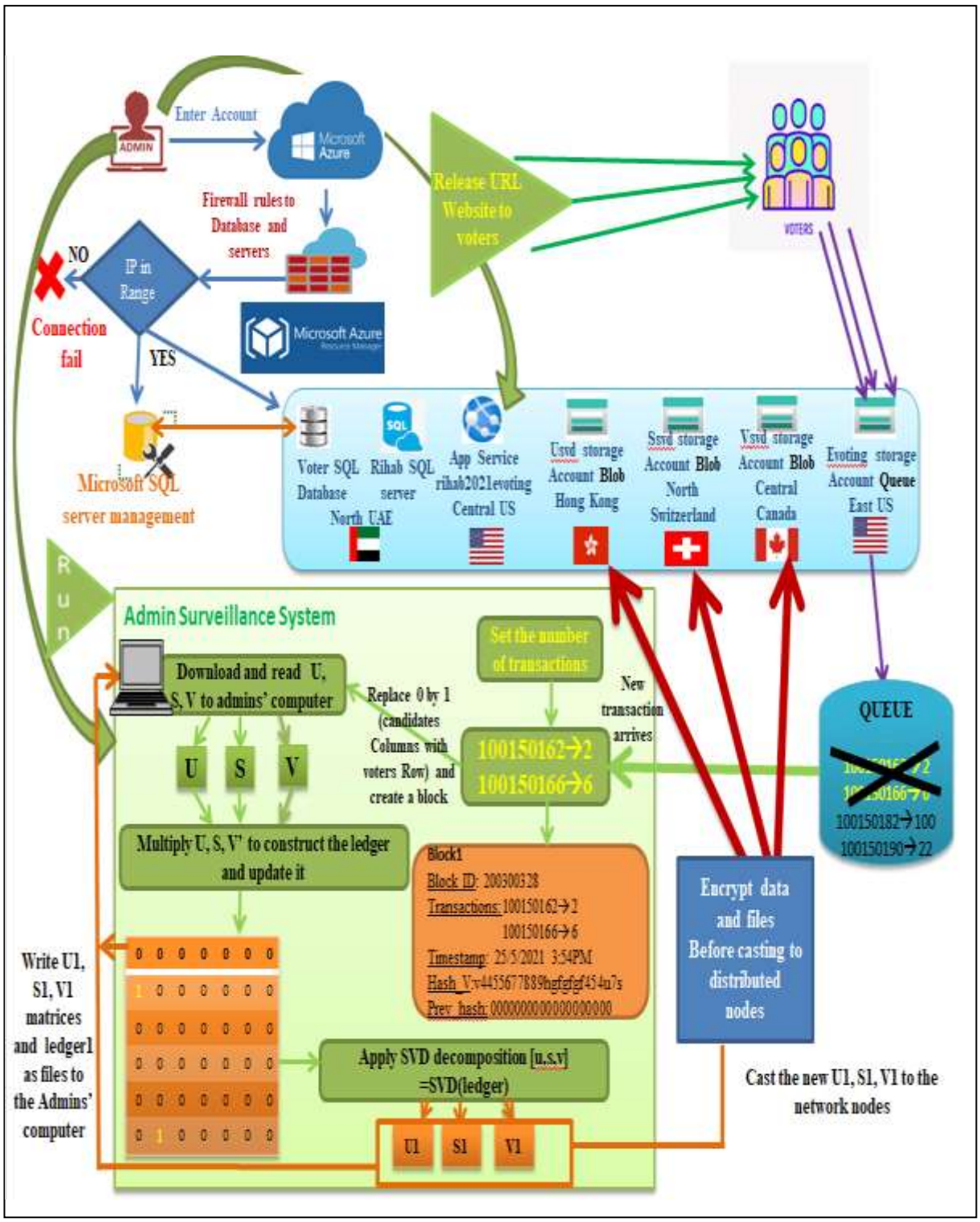


Figure (4-1). The topology of the proposed E-voting system

2- The resources include databases of the candidates and voters. A website application resource, three storage accounts for each output of SVD, and a storage account that holds the transactions temporarily.

3- After releasing the URL website application to the voters, the voting process begins and the votes are saved as transactions (Voter's ID → the desired candidate's ID).

4- The transactions of votes are fetched to the Admin surveillance system, in which the only capability is to enter the number of transactions for a block. The system keeps running automatically till the end of the election event.

5- As soon as the transactions arrive, a new block is constructed containing other information (Block ID, transactions, timestamp, a hash value, and the previous hash value for the previous block).

6- The ledger (in a form of a matrix) for the current block is updated to assign the value 1 for the location that corresponds to the voter's column and candidate's row.

7- SVD is applied on the ledger to construct the three SVD outputs (U, S, and V) which now contain the result in another form.

8- The three matrices and ledger for the current block are written as files and encrypted before casting them to distributed and separated network nodes.

9- The next block constructed, will download and read the three matrices from the distributed network nodes, and the ledger is updated by

adding the new votes to the previous ledger, producing incremental ledgers and updated copies of U, S, and V.

10- The final ledger and three matrices for the final block show the immediate results during the election event. This step can be performed to check the degree of integrity by the election committee.

4.3. Employing SVD in the E-voting System

The general ledger in the proposed e-voting system is of size 100*100000 (100 rows indicate the number of candidates and 100000 columns indicate the number of voters). Each time a block is created the SQL database for results stores the counts for candidates, at the same time SVD is applied on the general ledger that is updated adding the values of 1 for each citizens' vote corresponding to the row for a certain candidate. This process as mentioned previously aims to keep a copy of the results in another form that is distributed and for the final stage that is retrieved to match and check the rate of success for the whole event.

To show how SVD works in the proposed e-voting system, the following sections show how SVD works. Section 4.3.1. illustrates SVD for simplicity with an 8*10 ledger. Section 4.3.2 illustrates an actual example of SVD in the proposed e-voting system applied as 5 blocks each has 3 transactions.

4.3.1. SVD with an 8*10 Ledger

A ledger of 8*10 is an example where we have values of ones and zeros, in which each column has at most the value 1 referring to a vote for the desired candidate, the following ledger, as shown in table (4-1) is a copy of the SQL database for results. SVD works on this ledger to transform the values into another form and distribute the outputs in three storage nodes that are cast to different servers in the world.

*Table (4-1). General ledger of size 8*10.*

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
C1	0	0	1	0	0	0	0	0	0	1
C2	0	1	0	0	0	0	0	1	0	0
C3	0	0	0	0	0	1	0	0	0	0
C4	0	0	0	0	0	0	0	0	0	0
C5	0	0	0	1	0	0	0	0	0	0
C6	0	0	0	0	0	0	0	0	0	0
C7	0	0	0	0	1	0	0	0	0	0
C8	1	0	0	0	0	0	1	0	1	0

V: Voter, C: Candidate, Voter1 (V1) voted for candidate8 (C8), V2 voted for C2, and so on.

The validation of the ledger is first done by checking the values in which a voter has the right to vote only once so each column should have at most one value of 1.

The distributed nodes represent the outputs of SVD (U, S, and V matrices). The result of applying SVD for this example is shown in Table (4-2), Table (4-3), and Table (4-4):

*Table (4-2). The left matrix U_{8*8} for SVD is applied on a ledger 8*10.*

0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	-1
0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	0	-1	0	0	0
-1	0	0	0	0	0	0	0

*Table (4-3). The diagonal singular value matrix S_{6*6} for SVD is applied on a ledger 8*10.*

1.7321	0	0	0	0	0	0	0	0	0
0	1.4142	0	0	0	0	0	0	0	0
0	0	1.4124	0	0	0	0	0	0	0
0	0	0	1.0000	0	0	0	0	0	0
0	0	0	0	1.0000	0	0	0	0	0
0	0	0	0	0	1.0000	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

*Table (4-4). The right matrix V_{10*10} for SVD is applied on a ledger 8*10.*

-0.5774	0	0	0	0	0	0	0.5164	-0.1225	-0.6205
0	0.7071	0	0	0	0	-0.7071	0	0	0
0	0	0.7071	0	0	0	0	-0.5477	-0.0866	-0.4387
0	0	0	1.0000	0	0	0	0	0	0
0	0	0	0	-1.0000	0	0	0	0	0
0	0	0	0	0	1.0000	0	0	0	0
-0.5774	0	0	0	0	0	0	-0.2582	-0.6325	0.4472
0	0.7071	0	0	0	0	0.7071	0	0	0

-0.5774	0	0	0	0	0	0	-0.2582	0.7550	0.1733
0.0000	0	0.7071	0	0	0	0	0.5477	0.0866	0.4387

These matrices U, S, and V transformed the general ledger into coefficients of another form, producing an adaptive SVD ledger that is distributed in three separated countries using cloud storage services and servers.

Now we can deal with each matrix in a decomposed manner as the main aim of SVD is to reduce dimensionality on the general matrix without losing information. This is done by choosing a rank for the matrix depending on the singular values in matrix S (the diagonal matrix), in which we can use a low rank to deal with the three matrices more efficiently. Since the data is of binary type, which is suitable for elections, the rank can be as low as possible as long as it returns the exact matrix.

The rank of *r* or fewer results in an acceptable decomposition for large data sets, the following equation (4.1) is used to show how we deal with the size of large data sets of the binary form:

$$U(:,1:r) * S(1:r, 1:r) * V(:,1:r)' \quad (4.1)$$

Where we keep all the rows and only from 1 to *r* columns of U, all the rows and columns from 1 to *r* of S, and all the rows and only columns from 1 to *r* of V transpose.

So, instead of dealing with a ledger of size 80 (8*10) in this example, we can deal now with decomposed matrices of less size. For

example, if the rank of 3 is used, then we can deal with a matrix of size 63 (8*3 + 3*3 + 10*3) resulting from equation (4.2):

$$U*r + r*r + V*r \quad (4.2)$$

This result for massive binary matrices makes a huge difference compared with the original matrix. when retrieving the original matrix U*S*V' we get the same ledger with no loss of information as shown in Table (4-5) which is similar to the original general ledger shown in Table (4-6).

Table (4-5). Retrieving the original matrix.

0	0	1.0000	0	0	0	0	0	0	1.0000
0	1.0000	0	0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1.0000	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1.0000	0	0	0	0	0
1.0000	0	-0.0000	0	0	0	1.0000	0	1.0000	-0.0000

Table (4-6). The original matrix.

	V1	**V2**	**V3**	**V4**	**V5**	**V6**	**V7**	**V8**	**V9**	**V10**
C1	0	0	1	0	0	0	0	0	0	1
C2	0	1	0	0	0	0	0	1	0	0
C3	0	0	0	0	0	1	0	0	0	0
C4	0	0	0	0	0	0	0	0	0	0
C5	0	0	0	1	0	0	0	0	0	0
C6	0	0	0	0	0	0	0	0	0	0

C7	0	0	0	0	1	0	0	0	0	0
C8	1	0	0	0	0	0	1	0	1	0

4.3.2. SVD with 100*100000 Ledger of the E-voting System

The ledger of the proposed e-voting system consists of 100 candidates and 100000 voters. The number of transactions that are applied is 3 transactions and 5 blocks, where each block will contain 3 transactions as an example to show how SVD works in the proposed system, and how the transparency of the election event is measured.

As soon as the system begins to run, the general ledger and the three matrices U, S, and V are all of zero values, which we consider the genesis block.

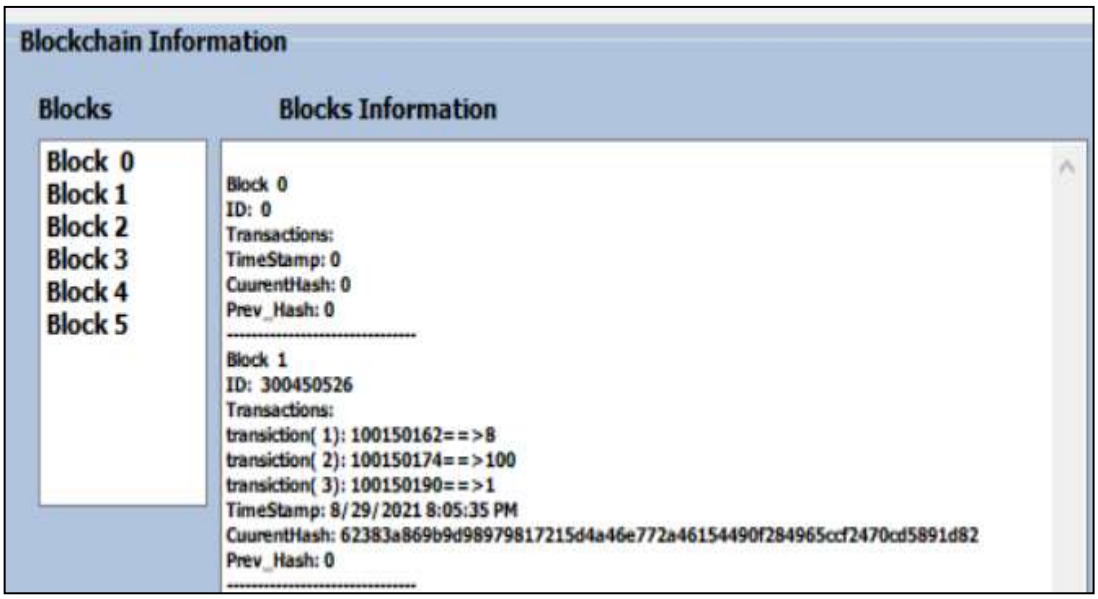
When the citizens log in to the e-voting web application and vote, the votes are stored as transactions temporarily in the queue storage to which the administrator system will keep listening.

Each block will contain a block ID which is the summation of all voters' IDs in the block, three transactions (votes) indicating the voter ID that voted for the candidate number, a timestamp indicating the date and time of the block creation, the hash value of the block and the hash value of the previous block.

The following scenario shows the ledgers and the three matrices for each block:

I. Block number one

Three transactions are fetched from the queue to construct the first block, the three transactions are the voters' IDs 100150162, 100150174, and 100150190 voted for candidate numbers 8, 100, and 1 respectively. Figure (4-2) shows the transactions and information in the genesis and the first block.



Figure(4-2). The block information contains the genesis and first block.

After creating the block the three matrices are downloaded, read, and construct the ledger by multiplying the three matrices U, S, and V transpose.

Then the new ledger is constructed according to the three transactions and updated, in which we can now extract the new U, S, and V for the first block by applying SVD decomposition.

The three matrices have a copy written as files to the administrators' computer and cast them to the distributed network nodes. All these steps are done and written to the administrators' platform one after the other as soon as each step is completed within less than a second as shown in Figure (4-3).

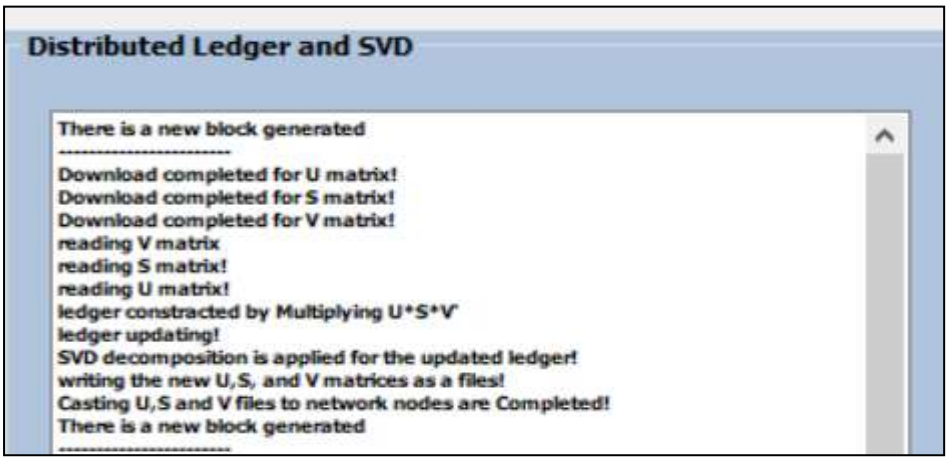


Figure (4-3). The performed steps for SVD on a block.

The ledger for block number 1 is shown in Figure (4-4). The ledger contains the votes for each voters' column corresponding to the candidates' row circled in red. In other words, the voter with ID 100150162 is the first voter in the database which means column number one correspond to candidate row number 8, a voter with ID 100150174 is the voter in column number 3 which is the third voter in the database that voted for candidate number 100 in row 100 and so on. The tables of the three matrices are shown in Figure (4-5) for the U, S, and V matrices respectively.

The tables concerned to the U and V matrix are cut to show the only first 40 rows of them. The actual tables contain 300 rows for matrix U and 3000 rows for matrix V.

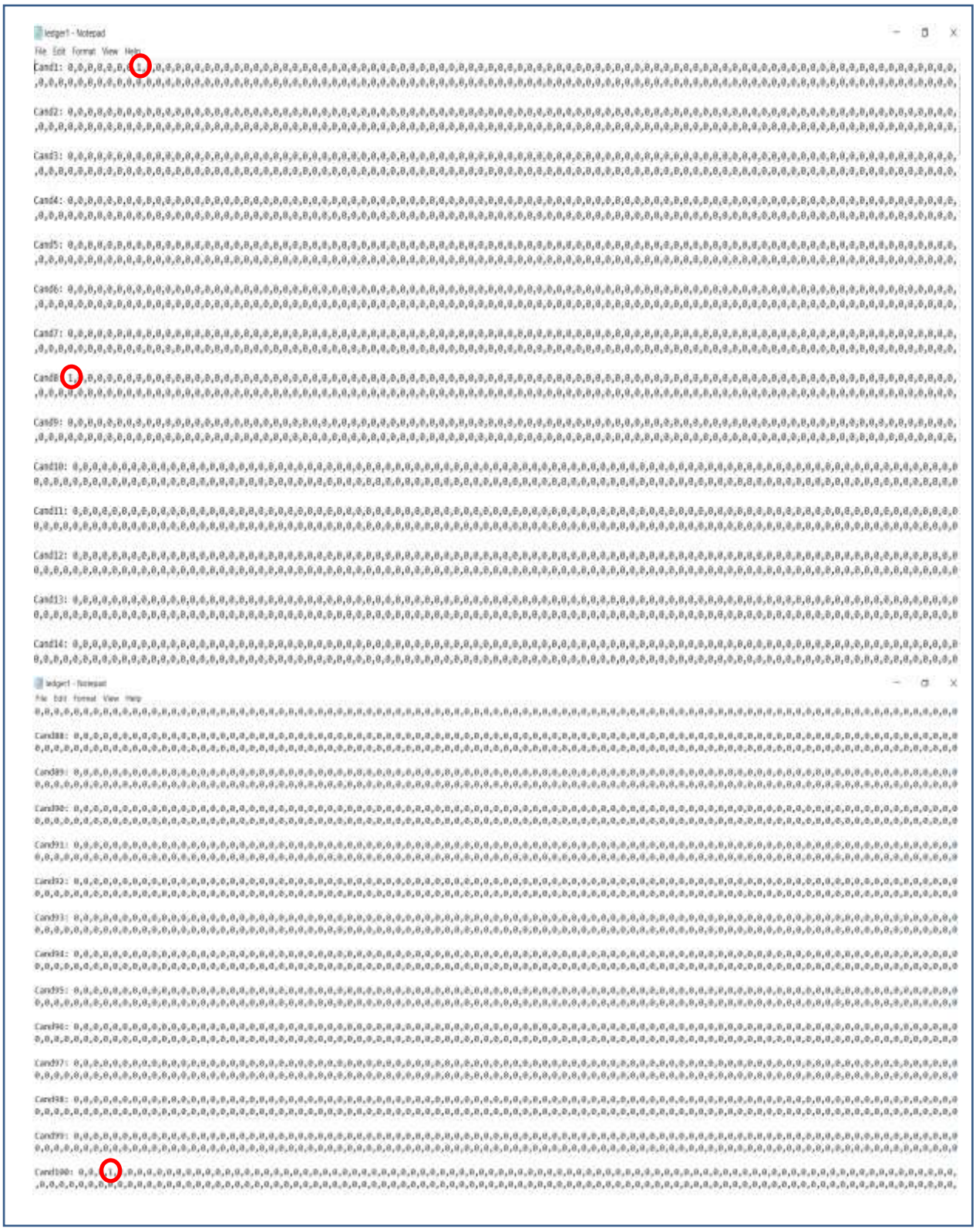


*Figure (4-4). **Ledger 1** for block number one.*

Matrix U1		
-1	0	0
0	0	0
0	0	0
0	0	0
-1	0	0
0	0	0
0	0	0
0	0	0
-1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Matrix S1		
1	0	0
0	0	0
0	0	0

Matrix V1		
0	0	-1
0	1	0
-1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Figure (4-5). Matrices U1, S1, and V1 for ***Block1****.*

II. Block number two

The three transactions that are fetched from the queue to construct the second block are the voters' IDs 100150214, 100150166, and 100150206 voted for candidate numbers 100, 5, and 77 respectively. Figure (4-6) shows the transactions and information for the second block.

Figure(4-6). Information for Block 2.

The ledger for block number 2 is shown in Figure (4-7). The ledger contains the votes for each voters' column corresponding to the candidates' row circled in green. In other words, the voter with ID 100150214 is the voter number 14 in the database which means column number 14 corresponds to the candidate row number 100, a voter with ID 100150166 is the voter number 2 in the database that voted for candidate row number 5 so on. The tables of the three matrices are shown in Figure (4-8) for the U, S, and V matrices respectively.

The tables concerned to the U and V matrix are cut to show the only first 40 rows of them. The actual tables contain 300 rows for matrix U and 3000 rows for matrix V.

Figure (4-7). **Ledger 2** for block number two.

Matrix U2		
-1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Matrix S2		
1	0	0
0	1	0
0	0	1

Matrix V2		
0	0	0
0	1	0
0	0	0
0	0	0
0	0	1
0	0	0
0	0	0
0	0	0
-1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Figure(4-8). Matrices U2, S2, and V2 for ***Block2***.

III. Block number three

The three transactions that are fetched from the queue to construct the third block are the voters' IDs 100150194, 100150210, and 100150202 voted for candidate numbers 2, 100, and 5 respectively. Figure (4-9) shows the transactions and information for the third block.

Figure(4-9). Information for Block 3.

The ledger for block number 3 is shown in Figure (4-10). The ledger contains the votes for each voters' column corresponding to the candidates' row circled in blue. In other words, the voter with ID 100150194 is the voter number 9 in the database which means column number 9 corresponds to the candidate row number 2, a voter with ID 100150210 is the voter number 13 in the database that voted for candidate row number 100 so on. The tables of the three matrices are shown in Figure (4-11) for the U, S, and V matrices respectively.

The tables concerned to the U and V matrix are cut to show the only first 40 rows of them. The actual tables contain 300 rows for matrix U and 3000 rows for matrix V.

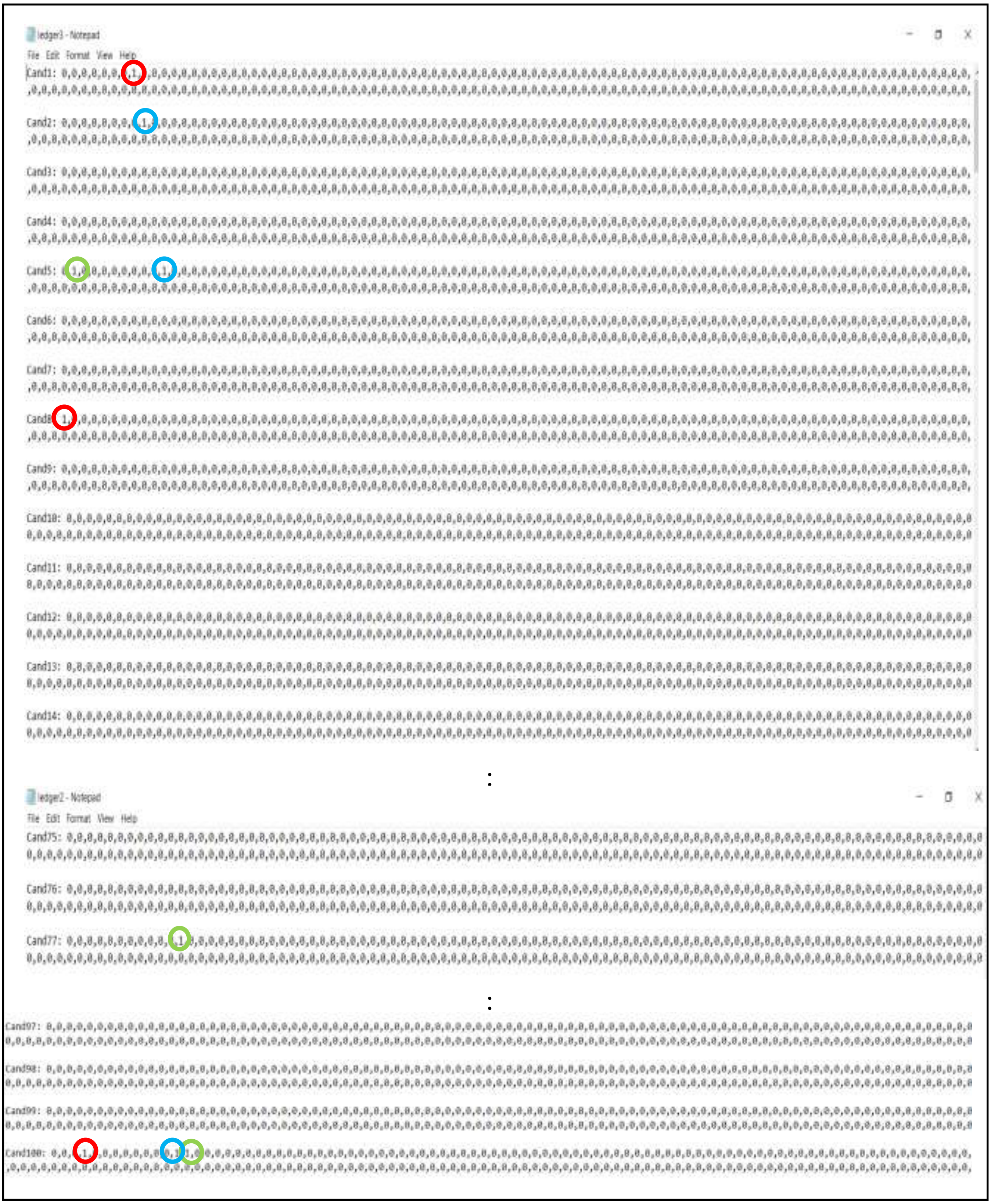


Figure (4-10). Ledger 3 for block number three.

Matrix U3		
0	0	0
-1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Matrix S3		
1.732051	0	0
0	1	0
0	0	1

Matrix V3		
0	0	0
0	0	1
0	0	0
0	0	0
-0.57735	0	0
0	0	0
0	0	0
0	0	0
0	-1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
-0.57735	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Figure(4-11). Matrices U3, S3, and V3 for ***Block3****.*

IV. Block number four

The three transactions that are fetched from the queue to construct the fourth block are the voters' IDs 100150182, 100150178, and 100150198 voted for candidate numbers 60, 2, and 55 respectively. Figure (4-12) shows the transactions and information for the fourth block.

Figure(4-12). Information for Block 4.

The ledger for block number 4 is shown in Figure (4-13). The ledger contains the votes for each voters' column corresponding to the candidates' row circled in purple. In other words, the voter with ID 100150182 is the voter number 6 in the database which means column number 6 corresponds to the candidate row number 60, a voter with ID 100150178 is the voter number 5 in the database that voted for candidate row number 2 so on. The tables of the three matrices are shown in Figure (4-14) for the U, S, and V matrices respectively.

The tables concerned to the U and V matrix are cut to show the only first 40 rows of them. The actual tables contain 300 rows for matrix U and 3000 rows for matrix V.

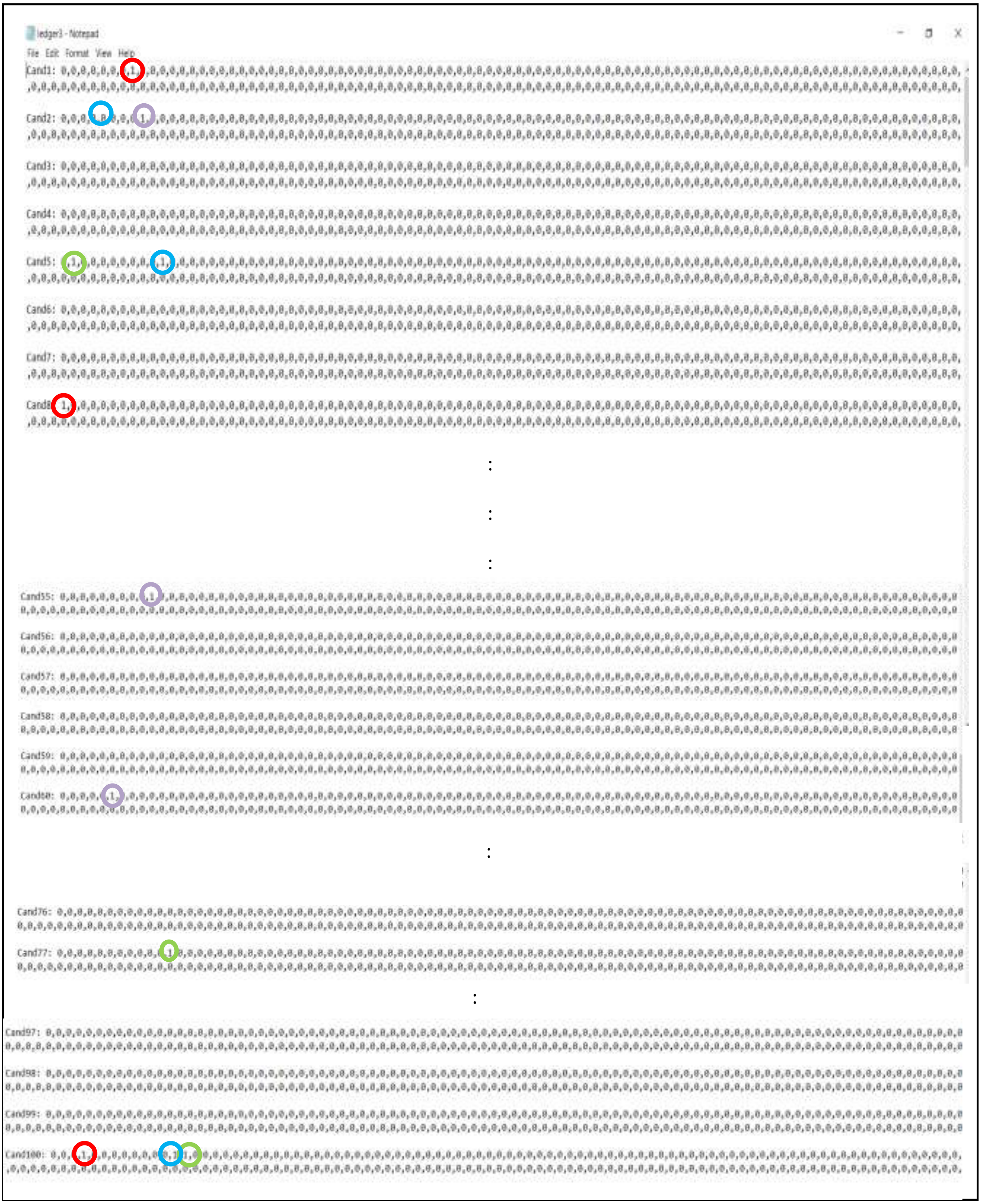


*Figure (4-13). **Ledger 4** for block number four.*

Matrix U4

0	0	0
0	0	1
0	0	0
0	0	0
-0.57735	0	0
0	0	0
0	0	0
0	0	0
0	-1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
-0.57735	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Matrix S4

1.732051	0	0
0	1.414214	0
0	0	1

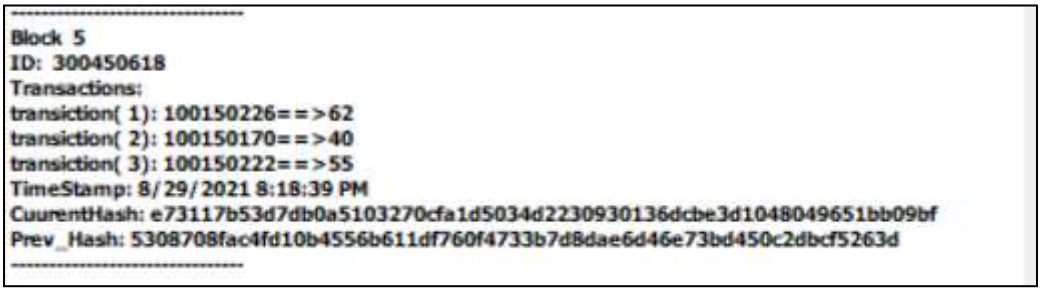
Matrix V4

0	0	0
0	0	0
0	-0.70711	0
0	0	0
-0.57735	0	0
0	0	0
0	0	0
0	0	0
0	0	-1
0	0	0
0	0	0
0	-0.70711	0
0	0	0
-0.57735	0	0
-0.57735	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Figure(4-14). Matrices U4, S4, and V4 for ***Block4****.*

V. Block number five

The three transactions that are fetched from the queue to construct the fifth block are the voters' IDs 100150226, 100150170, and 100150222 voted for candidate numbers 62, 40, and 55 respectively. Figure (4-15) shows the transactions and information for the fifth block.



Figure(4-15). Information for Block 5.

The ledger for block number 5 is shown in Figure (4-16). The ledger contains the votes for each voters' column corresponding to the candidates' row circled in brown. In other words, the voter with ID 100150226 is the voter number 17 in the database which means column number 17 corresponds to the candidate row number 62, a voter with ID 100150170 is the voter number 3 in the database that voted for candidate row number 40 and so on. The tables of the three matrices are shown in Figure (4-17) for the U, S, and V matrices respectively.

The tables concerned to the U and V matrix are cut to show the only first 40 rows of them. The actual tables contain 300 rows for matrix U and 3000 rows for matrix V.

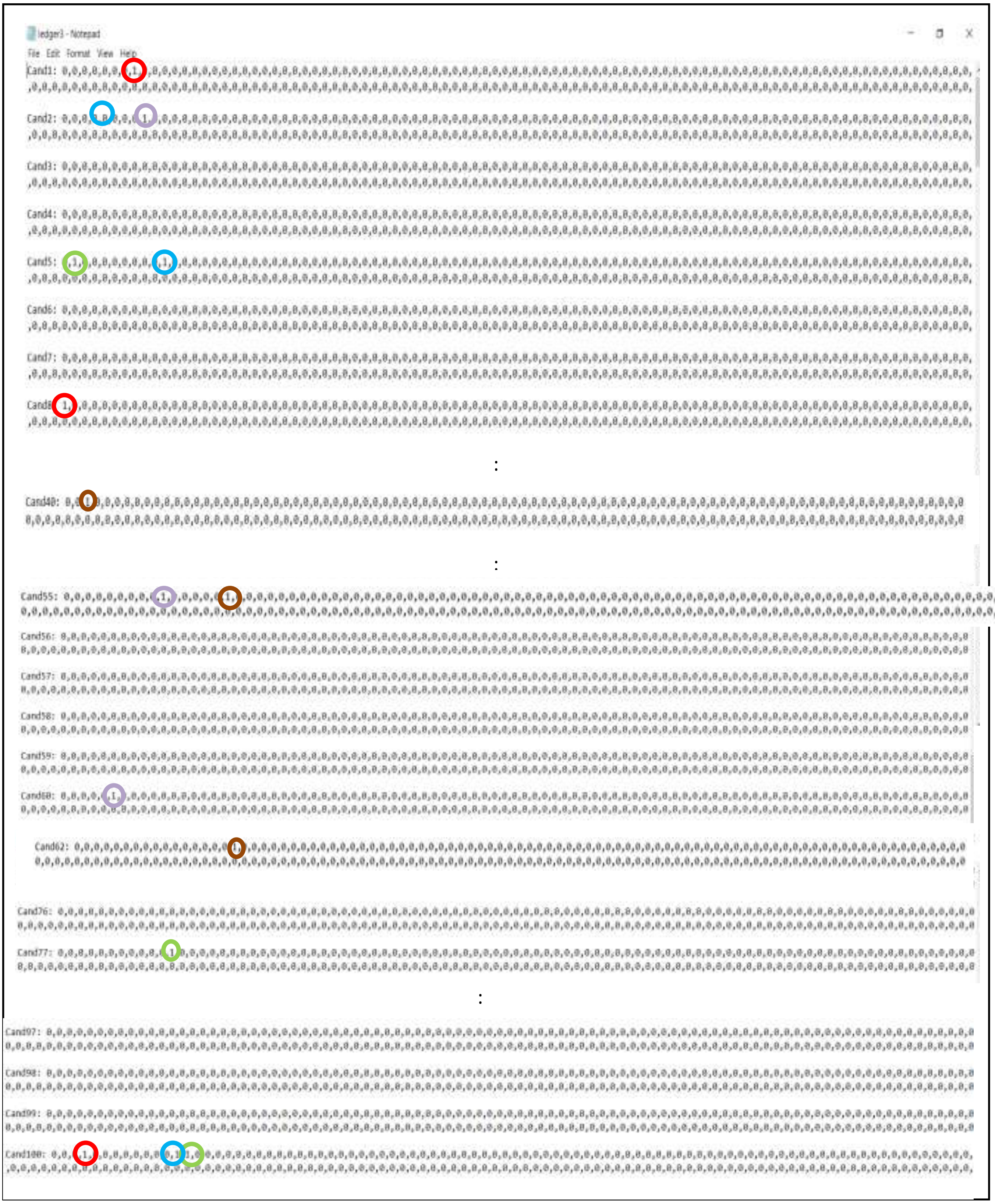


*Figure (4-16). **Ledger 5** for block number five.*

Matrix U5		
0	0	0
0	0	0
0	0	0
0	0	0
-1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
-1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Matrix S5		
1.732051	0	0
0	1.414214	0
0	0	1.414214

Matrix V5		
0	0	0
0	0	0
0	0	-0.70711
0	0	0
-0.57735	0	0
0	-0.70711	0
0	0	0
0	0	0
0	0	0
0	-0.70711	0
0	0	0
0	0	-0.70711
0	0	0
-0.57735	0	0
-0.57735	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Figure(4-17). Matrices U5, S5, and V5 for ***Block5***.

4.4. SVD with Binary Matrices

Employing singular value decomposition to Binary matrices can work efficiently, as data are exactly either 0 or 1. The reason is that instead of dealing with *m*n* of massive size, data of Boolean form can be dealt with SVD with rank $k \leq r$ in which r is the actual rank for a given matrix.

In special cases where data should contain at most only one 1 in a column under special circumstances, the rank can be as low as possible.

The following sections illustrate SVD in a special case on data for an e-voting system, and generally for matrices of binary form that can be used in all types of surveys, voting systems, symptoms of disease…etc.

4.4.1. Special Case of SVD with Matrices of Binary Data

The proposed e-voting system merges the concept of a distributed ledger with the Singular value decomposition technique for an e-voting system, in which a ledger contains transactions of votes where *m* represents the number of candidates, and *n* represents the number of voters.

SVD is applied on blocks of transactions (votes) where each block contains a fixed number of transactions. Under this circumstance, SVD manages data for each block to result in a ledger incrementally. In other words, SVD initially will be applied on a matrix of zero values every time a block of a few transactions arrives. Compared with the overall size of the matrix, SVD will need a low rank to work on each block incrementally.

If we have a matrix *A* of size *m*n*, where ***m<n***, under the following conditions:

1- Each column should have at most only one 1.
2- The data is treated as blocks of transactions, in which a few columns in the matrix are changed to achieve condition (1). So SVD will work on less data incrementally.

Then SVD can be applied efficiently by $r = k$, where r is the actual rank in which $r<m$ and k is the lowest rank based on the number of transactions in a block where $k \leq r$.

The size of the matrix that is retrieved after applying SVD is:
$m*k + k*k + n*k$ that is lower than the $m*n$.

4.4.2. General Case of SVD with Matrices of Binary Data

In the scenario where SVD is applied on all *m* and *n* at once for a matrix *A*, *r* is the actual rank, where:

$$r = \begin{cases} r < m & ,if\ m < n \\ r \leq n & ,if\ n < m \end{cases} \qquad (4.3)$$

whether *r* is the actual rank or less, in both ways it is acceptable to deal with the decomposed matrices instead of dealing with a binary matrix of massive size.

The experimental results are illustrated in Table (4-7) at the end of the section 4.4.2., comparing matrices of binary data once when $m<n$, another when $n<m$, and when $m=n$. Also, exhibiting the time needed for

generating a matrix under the condition of having one value of 1 randomly in each column all at once, the time needed for applying SVD, the suitable rank, the time needed for extracting back the matrix by multiplying $U*S*V^T$.

I. SVD for binary matrices where *M<N*

a. Binary matrix of size 10*100

Here we have *m=10,* and *n=100*

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 1.62 nanoseconds.
- The time needed for SVD to be applied on the original matrix is 222.38ns.
- The suitable rank *r* for retrieving the exact matrix is 8.
- The time needed after applying SVD and extracting the matrix is 11.73ns.

The significance lies in the time it takes for SVD to be applied, constructed, and extracts the matrix to match the original matrix, as shown in Figure (4-18).

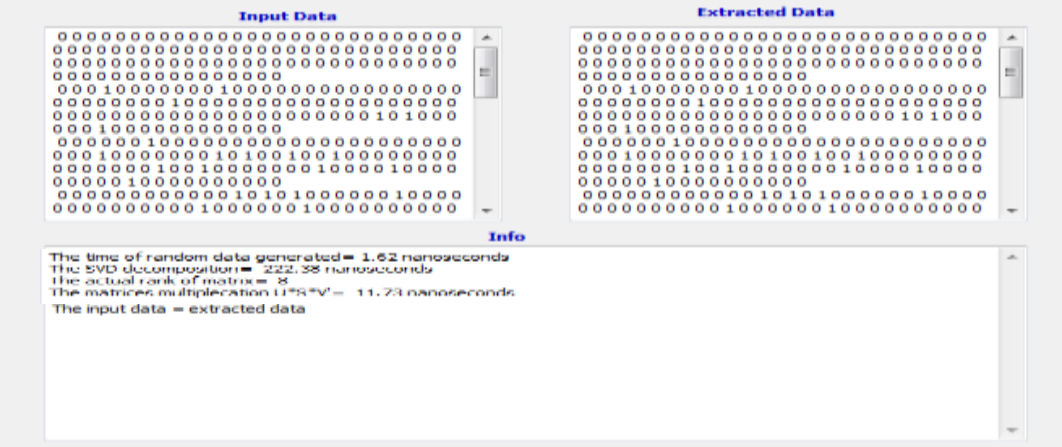


*Figure (4-18). Binary matrix of size 10*100 with rank 8.*

b. Binary matrix of size 50*100

m=50, and *n=100* where ***m<n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 0.63ns.
- The time needed for SVD to be applied on the original matrix is 201.45ns.
- A suitable rank *r* for retrieving the exact matrix is 40. However, after several tests, other ranks that retrieved the exact matrix are 39, 44, 42, 43, and 46 are shown in Table (4-7) at the end of section 4.4.2.
- The time needed after applying SVD and extracting the matrix is 205.62ns. Figure (4-19) shows the information practically.

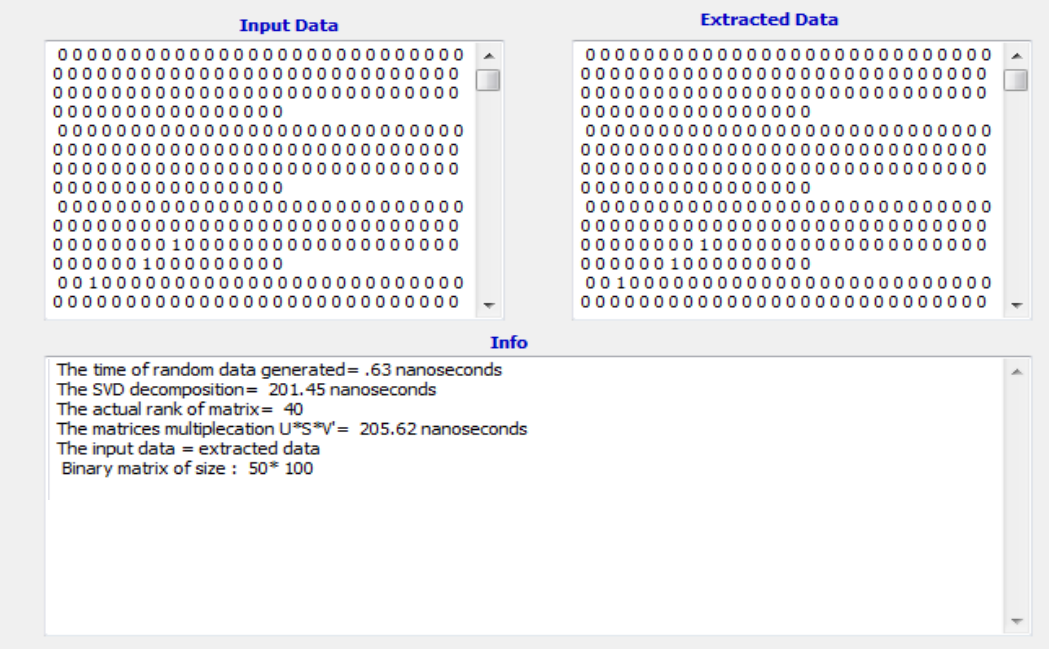


*Figure (4-19). Binary matrix of size 50 *100 with rank 40.*

c. Binary matrix of size 5*1000

We have *m=5,* and *n=1000* where ***m<n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 3.54ns.
- The time needed for SVD to be applied on the original matrix is 64.91ns.
- A suitable rank *r* for retrieving the exact matrix is 3.
- The time needed after applying SVD and extracting the matrix is 22.03ns. Figure (4-20) shows the information practically.

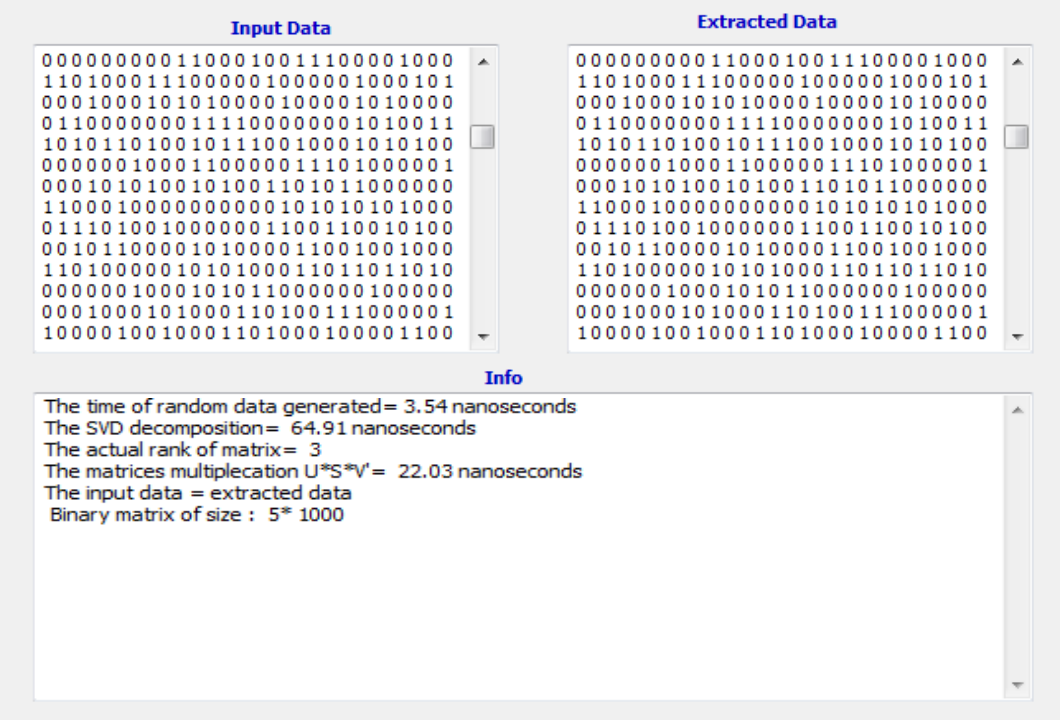


*Figure (4-20). Binary matrix of size 5*1000 with rank 3.*

d. Binary matrix of size 100*1000

We have *m=100,* and *n=1000* where ***m<n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 8.86ns.
- The time needed for SVD to be applied to the original matrix is 6.464µs.
- A suitable rank *r* for retrieving the exact matrix is 98.
- The time needed after applying SVD and extracting the matrix is 4.736 µs.

e. Binary matrix of size 100*100000

We have *m=100,* and *n=100000* where ***m<n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 758.59ns.
- The time needed for SVD to be applied on the original matrix is 1.9174ms.
- A suitable rank *r* for retrieving the exact matrix is 98.
- The time needed after applying SVD and extracting the matrix is 0.656ms.

II.SVD for binary matrices where *M>N*

a. Binary matrix of size 100*10

Here we have *m=100,* and *n=10* where ***m>n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 0.15ns.

- The time needed for SVD to be applied on the original matrix is 18.55ns.
- A suitable rank *r* for retrieving the exact matrix is 10. Though, after several tests, other ranks that retrieved the exact matrix are 7, 8, and 9 as shown in Table (4-7) at the end of section 4.4.2.
- The time needed after applying SVD and extracting the matrix is 12.43ns.

As long as the values of 1s are spread randomly in the binary matrix, it affects the value of the rank, in which $r \leq n$. So by repeating the test on this matrix we have different ranks that return the exact original matrix based on how the 1s are spread randomly due to the randomization function that is used for this task. Figure (4-21), shows the matrix 100*10 with a rank of 10.

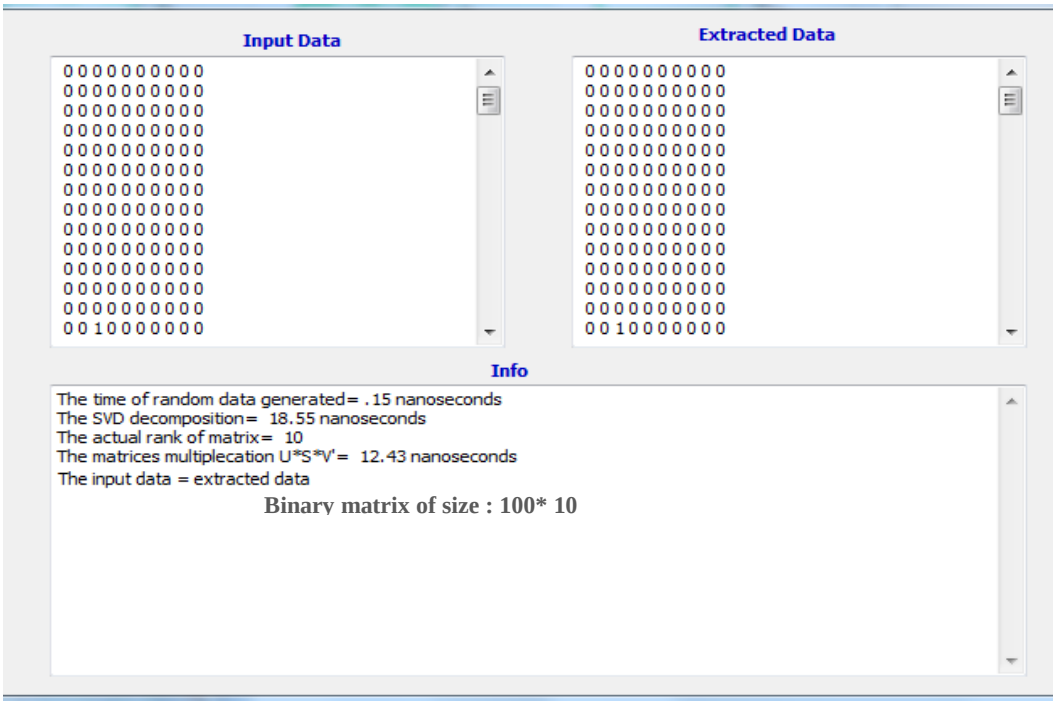


*Figure (4-21). Binary matrix of size 100*10 with rank 10.*

b. Binary matrix of size 100*50

Here we have *m=100,* and *n=50* where ***m>n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 0.49ns.
- The time needed for SVD to be applied on the original matrix is 271.78ns.
- A suitable rank *r* for retrieving the exact matrix is 34. However, after several tests, other ranks that retrieved the exact matrix are 36, 37, 38, 39, and 40 shown in Table (4-7) at the end of section 4.4.2.
- The time needed after applying SVD and extracting the matrix is 112.74ns. Figure (4-22) shows the information practically.

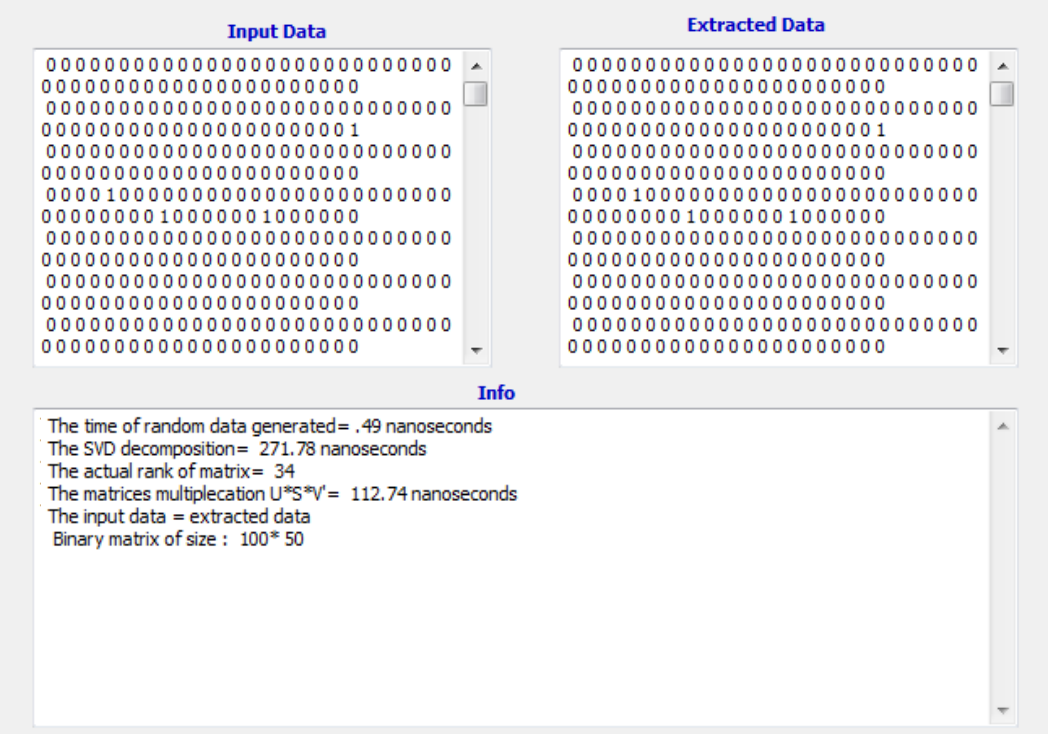


*Figure (4-22). Binary matrix of size 100 *50 with rank 34.*

c. Binary matrix of size 1000*5

Here we have *m=1000*, and *n=5* where ***m>n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 0.2ns.

- The time needed for SVD to be applied on the original matrix is 60.83ns.

- A suitable rank *r* for retrieving the exact matrix is 5.

-The time needed after applying SVD and extracting the matrix is 31.14ns. Figure (4-23) shows the information practically.

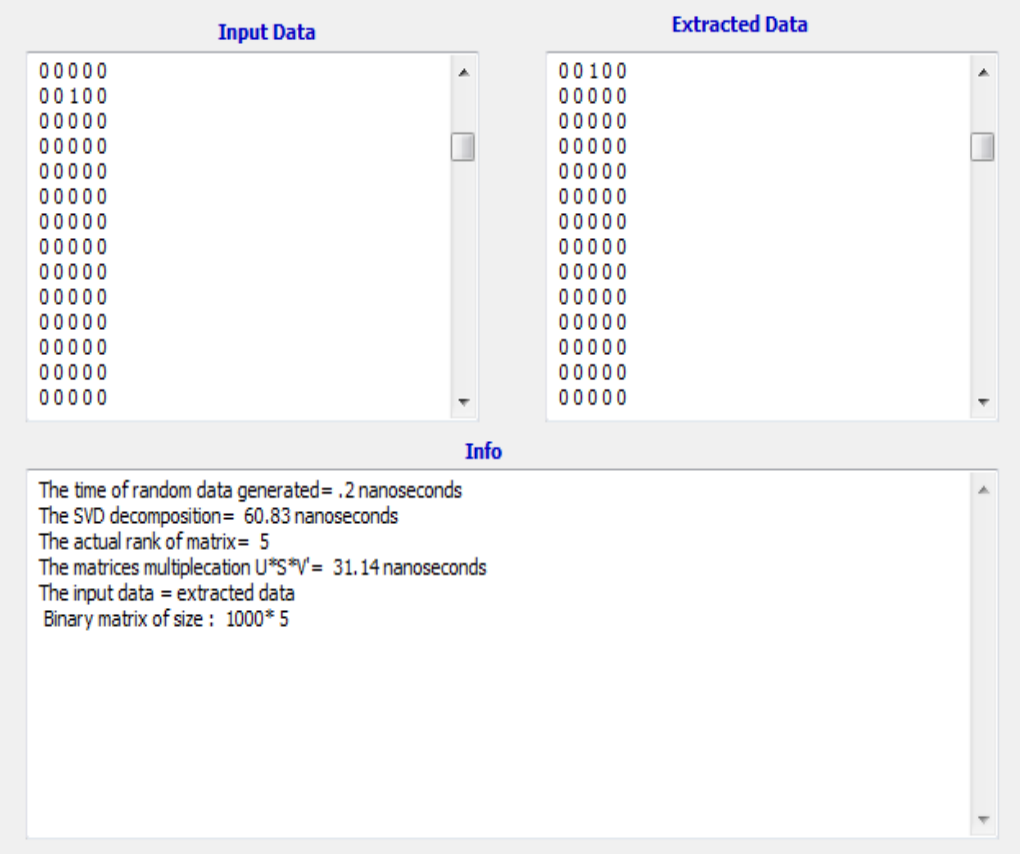


*Figure (4-23). Binary matrix of size 1000 *5 with rank 5.*

d. Binary matrix of size 1000*100

Here we have *m=1000,* and *n=100* where ***m>n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 3.9ns.
- The time needed for SVD to be applied on the original matrix is 5.5146µs.
- A suitable rank *r* for retrieving the exact matrix is 96. However, after several tests, other ranks that retrieved the exact matrix are 92, 95, 93, 97, and 91 shown in Table (4-7) at the end of section 4.4.2.
- The time needed after applying SVD and extracting the matrix is 4.1326 µs.

III.SVD for binary matrices where *M=N*

a. Binary matrix of size 100*100

Here we have *m* and *n=100* where ***m=n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 0.98ns.
- The time needed for SVD to be applied on the original matrix is 937.57ns.
- A suitable first rank *r* for retrieving the exact matrix is 67. Other ranks were 62, 64, 65, 60, and 63.

- The time needed after applying SVD and extracting the matrix is 746.34ns. Figure (4-24) shows the information practically.

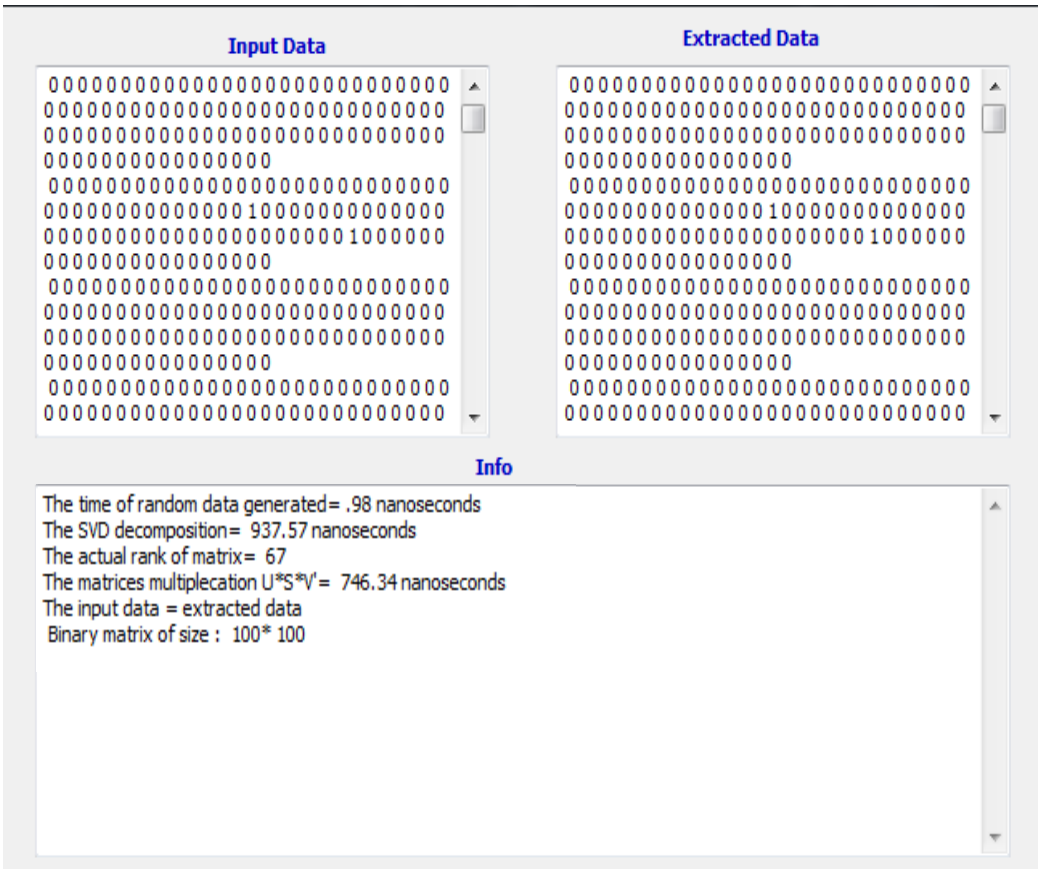


*Figure (4-24). Binary matrix of size 100*100 with rank 67.*

b. Binary matrix of size 500*500

Here we have *m* and *n=500* where ***m=n***.

- Before applying SVD, the time needed to generate the matrix using the randomization function to replace one zero by 1 randomly in each column is 11.5ns.
- The time needed for SVD to be applied on the original matrix is 55.565μs.
- A suitable first rank *r* for retrieving the exact matrix is 312. Other ranks were 307, 328, 314, and 329.

- The time needed after applying SVD and extracting the matrix is 38.174 µs.

The following table exhibits the results that retrieve the exact original matrix.

Table (4-7). Exhibiting the results of SVD with Binary Matrices.

	Size of the binary matrix M*N	**The time needed to generate the matrix**	**The time needed for SVD**	**The suitable ranks**	**Time for retrieving the matrix U*S*V'**
1	10*100	1.62ns	222.38ns*	8	11.73ns
2	50*100				
	Test1	0.99ns	203.28ns	39	224.33ns
	Test2	0.6ns	196.38ns	44	136.85ns
	Test3	0.76ns	352.3ns	42	232.05ns
	Test4	0.59ns	303.17ns	43	97.74ns
	Test5	0.45ns	140.52ns	46	299.8ns
	Test6	0.63ns	201.45ns	40	205.62ns
3	5*1000	3.54ns	64.91ns	3	22.03ns
4	100*1000	8.86ns	6.464µs*	98	4.736 µs
5	100*100000	758.59ns	1.9174ms*	98	0.656ms
6	100*10				
	Test1	1.17ns	18.57ns	7	10.31ns
	Test2	0.15ns	18.55ns	10	12.43ns

	Test3	0.11ns	20.75ns	9	11.34ns
	Test4	0.15ns	18.83ns	8	10.35ns
7	100*50				
	Test1	1.51ns	712.97ns	39	216.28ns
	Test2	0.49ns	299.53ns	36	212.69ns
	Test3	0.51ns	309.82ns	40	232.4ns
	Test4	0.47ns	315.33ns	37	190.27ns
	Test5	0.49ns	271.78ns	34	112.74ns
	Test6	0.48ns	231.24ns	38	211.83ns
8	1000*5	0.2ns	60.83ns	5	31.14ns
9	1000*100				
	Test1	3.9ns	5.5146µs	96	4.1326 µs
	Test2	2.63ns	5.1909µs	92	4.2357 µs
	Test3	2.9ns	5.3805µs	95	4.4833 µs
	Test4	1.85ns	5.5251µs	93	4.1490 µs
	Test5	2.96ns	5.5490µs	97	4.3561 µs
	Test6	2.78ns	5.4024µs	91	4.10464 µs
10	100*100				
	Test1	0.98ns	937.57ns	67	746.34ns
	Test2	0.93ns	951.27ns	62	514.8ns
	Test3	0.95ns	579.82ns	64	631.9ns
	Test4	0.85ns	613.66ns	65	475.99ns

	Test5	0.92ns	835.36ns	60	633.98ns
	Test6	0.91ns	652.66ns	63	711.19ns
11	500*500				
	Test1	11.5ns	55.565µs	312	38.174 µs
	Test2	15.83ns	55.241µs	307	37.788 µs
	Test3	21.72ns	55.011µs	328	41.464 µs
	Test4	12.9ns	5.543µs	314	39.766 µs
	Test5	11.79ns	57.305µs	329	42.773 µs

*ns: nanosecond, µs: microsecond, ms: millisecond

4.5. Properties of SVD with Binary Matrices

Singular value decomposition with rectangular binary matrices where $m<n$ can have rank $r<m$ in which r is less than m to extract the exact or approximate matrix. In special cases where SVD is employed within distributed ledger in which the ledger of rectangular binary data is constructed incrementally, the rank can be less than m because it deals progressively with a low amount of data compared with the whole massive size of A incrementally depending on the number of transactions in each block.

When $m>n$ the rank is $r\leq n$ in which r can be at most equal or no more than n to extract the exact or approximate matrix.

In Binary matrices, whether the matrix is of rectangular or square data, the rank is automatically chosen based on the smallest number of rows *n* or columns *m.*

According to the experimental results, applying SVD on binary matrices with massive size does not exceed even one second, which makes it useful in applications for such types of data.

Whenever the matrix is large and of binary form it can be decomposed according to the rank that can be less or equal to *n or m* to retrieve or cast matrices in a decomposed manner instead of matrices with size *n*m*. The experimental results show that whenever *n* or *m* is much smaller than each other, the chances of having several ranks to extract the exact matrix are very low. Also, it is shown that SVD with rectangular binary matrices takes less time than square binary matrices to be applied and extracted.

All these processes done by SVD allows us to employ this technique not only for the decomposition reason that allows us to save the matrices in a distributed manner to avoid locality but also as a matching tool that retrieves the results in another forum where we produce a ledger of results and match the similarity of results rating the integrity of an election event.

4.6. Measuring the Transparency of Election

Employing the SVD technique will not only aim to reduce the dimensionality of the ledger but also speed up the process for casting

and retrieving the ledger in less amount time. Also, this ledger that is managed by SVD is used at the end of the election to be matched with the results. If they match, then the whole election event is successful. In case an attack such as SQL injection occurs (where the hacker is capable to request records and change the values in the SQL database), then we depend on the last SVD ledger for the right final result. In other words, the adaptive SVD ledger is employed as a matching tool that is distributed and stored in three different places for security reasons, this adaptive ledger process the data and transforms it immediately as soon as the votes arrive as transactions in each block.

A part of the online election results that are matched with the final ledger is shown in table (4-8). The whole results are shown in appendix (B). In which a count to the number of 1s in the final ledger is matched with the number of votes (n_votes) on the result page.

According to the final ledger in Figure (4-16) and the results in table (4-8), we have:

- Candidate number 1 has 1 vote.
- Candidate number 2 has 2 votes.
- Candidate number 5 has 2 votes.
- Candidate number 8 has 1 vote.
- Candidate number 40 has 1 vote.
- Candidate number 55 has 2 votes.
- Candidate number 60 has 1 vote.
- Candidate number 62 has 1 vote.
- Candidate number 77 has 1 vote.
- Candidate number 100 has 3 votes.

Table (4-8). The online election results that match the counts in the final ledger.

ID	Candidate Name	N. of Votes	Percentage
1	علي محمد حسين	1	6
2	محمد حسن عبيد	2	13
3	غفران جاسم كاظم	0	0
4	احمد رحيم كريم	0	0
5	علي عبد الرحمن	2	13
6	رغدة عبد علي	0	0
7	شاكر جواد فاضل	0	0
8	شهلاء عبد القادر	1	6
9	سعد لطيف عبد	0	0
10	قاسم عبد الكريم	0	0
11	رسول مجيد حميد	0	0
12	نهى جبار ناهي	0	0
13	غفار عبد الله محمد	0	0
14	تقى هادي مهدي	0	0
15	رنا عبد الاله	0	0
16	ضمياء حسن علي	0	0
17	منى عبد الرزاق	0	0
18	لمياء قصي باقر	0	0
19	حميدة عبد الامير	0	0
20	بسام فاضل عباس	0	0
21	عباس عبد القاسم	0	0
22	ناجي هاني مسعود	0	0
23	مصطفى مهدي محمود	0	0
24	حسن حسين ستار	0	0

4.7. Evaluating the E-voting System

The E-voting system is evaluated from different sides of view as follows:

4.7.1. Security View

The e-voting system has several levels of security:

1- Votes are secured within a block structure using the SHA256 hash function.

2- Opening a connection between the system and the cloud is secured by firewall rules.

3- Databases, files, servers, and other resources connected to the cloud services are secured by the AES256 cryptograph algorithm.

4- The overall system cannot be controlled by any person as soon as the election event begins.

4.7.2. Transparency View

The systems' transparency is evaluated through a mechanism that aims to store another copy of the results immediately as soon as a block is created. This mechanism is done by SVD as follows:

1- At the beginning, the ledger and the matrices U, S, and V are reset to zero values.

2- SVD straightway is applied on each block of votes to construct and update a ledger and the three outputs of SVD (U, S, V singular matrices) for that block.

3- The three matrices and ledger that are updated are downloaded as files and cast to the network nodes in three separated servers.

4- Each time a ledger is updated, it contains information on the previous ledger. In which there exist copies of an incremental ledger and its matrices U, S, and V.

5- The transparency of the election event is measured by matching the count of votes in the final ledger with the number of votes in the SQL database of results.

6- The system keeps copies for every block's ledger and the three SVD matrices in the system and in the remote distributed network nodes that can be checked by following the data.

Chapter Five

Conclusion and Future Work

5.1. Introduction

This chapter introduces conclusions on what ideas can be adapted to employ the singular value decomposition (SVD), especially when the data of the matrix is in a binary form, and what are the main developing features are used. Also, this chapter introduces future works that can be solved later.

In this chapter, section 5.2 is the developed system conclusions, and section 5.3 is the future works that are recommended in some fields of research.

5.2. Conclusion

Through the design, implementation, and discussion of the results of the system, the following concluded essential remarks:

1- The online e-voting system (web application) can count the results immediately.

2- The system works dynamically in real-time and ends under a certain condition which is the time set for the election event to end.

3- The system employs SVD as a matching tool that constructs the ledger matrix of results during the election event, then compares the counts for each candidate in the final ledger with the online election results (SQL database of results) rating the integrity of the election event.

4- The system employs SVD for dimensionality and data reduction, to deal with low dimensionality data without losing the important data saving time and space.

5- The system employs SVD to distribute its results to separated network nodes in different servers for security reasons.

6- Storing the output of SVD (U, S, V^T) in separated servers, is a way to add a degree of security, as the important components of the system are not locally managed or saved.

7- The Singular Value Decomposition technique is a recommended tool that works efficiently with datasets of binary data.

8- Incremented ledgers are produced for every block containing transactions of votes which can be used to check the rate of integrity for the whole election process at the end of the event.

9- The system secures the transaction of votes in a block structure secured by the hash function (SHA 256) that is applied on the ID of each block (which is the sum of all voters' IDs in that certain block). Each block is linked to the previous block by this hash function.

10- The proposed e-voting system achieved an acceptable degree of transparency, security, integrity, uniqueness, verifiability, anonymity to keep the system trusted and as safe as possible.

5.3. Future Work

The suggestions of the future works are as follows:

1- The system can be improved by using Biometric algorithms within the confirmation phase as a powerful authentication tool, such as face recognition algorithms, fingerprints, eye retina…etc.

2- The SVD for binary matrices is a suitable tool for many applications, such as election applications of any type, surveys for organizations and companies, educational surveys, healthcare applications indicating symptoms of disease…etc.

3- Security algorithms can be employed within cloud security services, or within the administrators' back end, to create a more robust system against specific attacks.

4- The blocks of transactions (votes) can be distributed between all or some nodes in the network, adding a more powerful degree of validation for the election results.

5- The IDs of voters can be more anonymous by applying a hash function that encrypts the identity of voters.

6- The distributed network nodes (U, S, and V) that contain the election results (processed by SVD), can be shared to all main nodes in the network to obtain a more robust degree of verifiability.

References

References

[1] Marco Prandini, Laura Sartori and Anne-Marie Oostveen, "Why electronic voting", International Conference for e-Democracy and Open Government, Hong Kong, 2014.

[2] Charles K., J.O. Daramola, and A. A. Azeta, "Developing a Secure Integrated E-Voting System", Chapter in a Handbook of Research on E-Services in the Public Sector: E-Government Strategies and Advancements, Published in the United States of America by Information Science Reference, pp. 278-287, 2011.

[3] Friðrik Þ Hjálmarsson and Gunnlaugur K Hreiðarsson, "Blockchain-based e-voting system", IEEE International Conference on Cloud Computing, USA, 2018.

[4] Aishwarya Indapwar, Manoj Chandak, and Amit Jain, "E-voting system using Blockchain technology", International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 3, pp. 2775-2779, 2020

[5] Faraz Masood and Arman Rasool Faridi, "An Overview of Distributed Ledger Technology and its Applications", International Journal of Computer Sciences and Engineering, vol. 6, no. 10, pp. 422-427, 2018.

[6] Yasmine M. Tabra, "Internet voting system in Iraq", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 10, pp. 47-52, 2013

[7] Zihan Chen, "Singular Value Decomposition and its Applications in Image Processing", International Conference on Mathematics and Statistics, pp. 16- 22, 2018

[8] Jure Leskovec, Anand Rajaraman, and Jeff Ullman, "Mining of massive datasets", Cambridge University Press, 2nd edition, 2014.

[9] Aakash S., Aashish, Akshit, and Sarthak, "Online Voting system", SSRN electronic library, pp. 1-5, 2020.

References

[10] G.Kalaiyarasi, K. Balaji, T. Narmadha, and V. Naveen, "E-Voting System In Smart Phone Using Mobile Application", International Conference on Advanced Computing and Communication Systems, pp. 1466- 1469, Coimbatore, India, 2020.

[11] Ahmed Ben Ayed, "A Conceptual Secure Blockchain-Based Electronic Voting System", International Journal of Network Security and Its Applications, vol. 9, no.3, pp. 5-9, 2017.

[12] Ramya Govindaraj, Kumaresan P and K.Sree Harshika, "Online Voting System using Cloud", International Conference on Emerging Trends in Information Technology and Engineering, pp. 1-4, Vellore, India, 2020

[13] S Sekar, C Vigneshwar, J Thiyagarajan, V B Soorya Narayanan and M Vijay, "Decentralized e-voting system using Blockchain", International Research Journal of Engineering and Technology, vol.7, no. 03, pp. 312-324, 2020.

[14] Michał Pawlak, Aneta Poniszewska-Maranda and Natalia Kryvinska, "Towards the intelligent agents for blockchain e-voting system", The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks, pp. 239–246, 2018.

[15] Denise Demirel, Richard Frankland, and Melanie Volkamer, "Readiness of various eVoting systems for complex Elections", Technical report, 2011.

[16] Elham Akbari, "From Blockchain to Internet-Based Voting", Master thesis, Department of Electrical Engineering and Computer Science, Cleveland State University, Ohio, US, 2018.

[17] Andrew Barnes, Christopher Brake, and Thomas Perry, "Digital Voting with the use of Blockchain Technology", Plymouth University, England, UK, 2017.

[18] Yifan Wu, "An E-voting System based on Blockchain and Ring Signature", Master thesis, School of Computer Science, University of Birmingham, England, UK, 2017.

References

[19] Frank Emmert, Antony page and Christopher page, "Trouble Counting Votes, Comparing Voting Mechanisms in the United States and Selected Other Countries", Creighton Law Review, vol. 41, 2007.

[20] Stavros Valsamidis, Sotirios Kontogiannis, Theodosios G Theodosiou and Ioannis Petasakis, " Web e-voting system with a data analysis component", Journal of Systems and Information Technology, vol. 20, no. 1, pp. 33-53, 2018.

[21] Madhuri Namballa, Mend Vaishnavi, Tejasree Kaka, Duvvuri Sai Suma, and K. Sriram, "Realtime Fingerprint-based Voting System", International Journal of Engineering Research and Technology, vol. 9, no. 09, pp. 59-62, 2020.

[22] Francesco Fusco, Maria Ilaria Lunesu, Filippo Eros Pani and Andrea Pinna, "Crypto-Voting, a blockchain-based e-voting system", 10th International Conference on Knowledge Management and Information Sharing, vol. 3, pp. 223-227, Seville, Spain, 2018.

[23] Website: "Fingerprint Recognition Based Biometric Voting Machine", accessed on 20 July 2021, https://www.edgefx.in/fingerprint-recognition-based-biometric-voting-machine/

[24] Rifa Hanifatunnisa and Budi Rahardjo, "Blockchain-based e-voting recording system design", 11th International Conference on Telecommunication Systems Services and Applications, pp. 1-6, Lombok, Indonesia, 2017.

[25] Haibo Yi, "Securing e-voting based on blockchain in P2P network" Journal on Wireless Communications and Networking, 137, pp. 1-9, 2019.

[26] Ishaku Liti Awalu, Park Hung Kook and Joa San Lim, "Development of a Distributed Blockchain e-voting System", Proceedings of the 10th International Conference on E-business, Management and Economics, pp. 207–216, Beijing, China, 2019.

References

[27] Yousif Abuidris, Abdelrahman Hassan, Abdalla Hadabi, and Issameldeen Elfadul, "Risks and Opportunities of Blockchain-Based on E-Voting Systems", 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, pp. 365- 368, Chengdu, China, 2014.

[28] M. Campbell-Verduyn and M. Goguen, "Blockchains, trust and action nets: extending the pathologies of financial globalization", Global Networks, vol. 19, no. 3, pp. 308-328, 2019.

[29] H. D. Clarke, M. Goodwin, and P. Whiteley, "Why Britain voted for Brexit: an individual-level analysis of the 2016 referendum vote", Parliamentary Affairs, vol. 70, no. 3, pp. 439-464, 2017.

[30] Chetan Sontakke, Swapnil Paygha, Shivkumar Raut, Shubham Deshmukh, Mayuresh Chande, and D. J. Manowar, "Online Voting System via Mobile", International Journal of Engineering Science and Computing, vol. 7, no. 5, pp. 12176-12178, 2017.

[31] Preeti Ahlawat and Rainu Nandal, "Performance Improvement using Pseudorandom One Time Password (OTP) in Online Voting System", Journal of Computer Engineering, vol. 17, no. 5, pp. 31-38, 2015.

[32] Mohit Sharma and Manisha Nene, "Quantum One Time Password With Biometrics", Data Engineering and Communications Technologies series, Chapter: 37, Springer Nature, 2019.

[33] Dhiraj P. Girase, "A Secure Smartphone-Based Voting System with Modified EVM Using Elliptic Curve Cryptography", International Journal of Electronics and Communication Engineering, vol. 8, no. 1, pp. 91-98, 2015.

[34] Shaan R., (2018), "The Difference Between Blockchains & Distributed Ledger Technology", Retrieved March 22, 2019, from Towards Data Science website: https://towardsdatascience.com/the-difference-between-

blockchains-distributedledger-technology-42715a0fa92

[35] Curran K., "E-Voting on the Blockchain", The Journal of the British Blockchain Association, vol. 1, no. 2, pp. 1–6, 2018.

[36] T. P. Abayomi-Zannu, I. A. Odun-Ayo, and T. F. Barka, "A Proposed Mobile Voting Framework Utilizing Blockchain Technology and Multi-Factor Authentication", International Conference on Engineering for Sustainable World, Journal of Physics: Conference Series, Vol. 1378, pp. 1-8, 2019.

[37] Website: Geeks for Geeks, "Decentralized Voting System using Blockchain", accessed on 8 August 2021, https://www.geeksforgeeks.org/decentralized-voting-system-using-blockchain/

[38] Rihab H Sahib and Eman S. Al-Shamery, "A Review on Distributed Blockchain Technology for E-voting Systems", International Conference of Modern Applications on Information and Communication Technology (ICMAICT), October 2020, Journal of Physics: Conference Series vol. 1804, pp. 1-23, 2021.

[39] Fran Casino, Thomas K.Dasaklis and Constantinos Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification, and open issues", Telematics and Informatics, vol. 36, pp. 55-81, 2019.

[40] Ruhi Tas and Ömer Özgür Tanrıöver, "A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting", Molecular Diversity Preservation International Symmetry Journal, vol. 12, no. 8, 2020.

[41] Mischa Tripoli and Josef Schmidhuber, "Emerging Opportunities for the Application of Blockchain in the Agrifood Industry", Published by Food and Agriculture Organization of the United Nations and International Centre for Trade and Sustainable Development (ICTSD), Switzerland, 2018.

[42] Imran Bashir, "Mastering Blockchain", Second Edition, Packt Publishing Ltd, Birmingham, UK., 2018.

References

[43] Michel Rauchs, Andrew Glidden, Brian Gordon, Gina Pieters, Martino Recanatini, François Rostand, Kathryn Vagneur, and Bryan Zhang, "Distributed Ledger Technology Systems: A Conceptual Framework", University of Cambridge, England, UK, 2018.

[44] International Telecommunication Union ITU-T, Technical Report FG DLT D1.2, "Distributed Ledger Technology, overview, concepts and ecosystem", 2019.

[45] Duneesha Fernando and Nalin Ranasinghe, "Permissioned Distributed Ledgers for Land Transactions; A Case Study", In book: Business Process Management: Blockchain and Central and Eastern Europe Forum, BPM Blockchain and CEE Forum Vienna, Austria Proceedings pp.136-150, Austria, 2019.

[46] Harsh Desai, Murat Kantarcioglu and Lalana Kagal, "A Hybrid Blockchain Architecture for Privacy-Enabled and Accountable Auctions", IEEE International Conference on Blockchain, pp. 34-43, Atlanta, GA, USA, 2019.

[47] Ali Sunyaev, "Distributed Ledger Technology", Chapter 9 In book: Internet Computing, pp. 265-299, 2020.

[48] Fred Douglis and Angelos Stavrou, "Distributed Ledger Technologies", IEEE Internet Computing, pp.5-6, 2020.

[49] Nabil El Ioini and Claus Pahl ,"A Review of Distributed Ledger Technologies", OTM Conferences - Cloud and Trusted Computing, pp. 1-13, 2018.

[50] Jen-Ho Hsiao1(&) , Raylin Tso1 , Chien-Ming Chen2 , and Mu-En Wu3, "Decentralized E-Voting Systems Based on the Blockchain Technology", In book: Advances in Computer Science and Ubiquitous Computing, Springer Nature Singapore Pte Ltd., pp.305-309, 2018.

[51] Eugenia Politou, Fran Casino and Constantinos Patsakis "Blockchain Mutability: Challenges and Proposed Solutions", arXiv:1907.07099v1 [cs.CR], 16 Jul 2019.

[52] Jasmeet Kaur and Jyotsn: "A Review of Blockchain Technology in Education", A Journal of Composition Theory, vol. 13, no. 4, pp. 392-400, 2020.

[53] Emanuela Srbinovska and Pece Mitrevski, "Web Services Deployment in Microsoft Azure Cloud Computing Platform", International Scientific Conference on Information, Communication and Energy Systems and Technologies, vol. 1, pp. 75-78, Serbia, 2014.

[54] Website: "Teleco advises business to move to the Cloud with MS Azure", accessed 16 April ,2021, https://teleco.ca/teleco-advises-business-to-move-to-the-cloud-with-ms-azure/

[55] G. Carutasu, M. A. Botezatu, C. Botezatu and M. Pirnau, "Cloud Computing and Windows Azure", International Conference 8th Edition Electronics, Computers and Artificial Intelligence, pp. 7-12, Ploiesti, Romania, 2016.

[56] T. Dykstra, R. Anderson and M. Wasson, "Building Real World Cloud Apps With Windows Azure", Microsoft, E-book, 2014.

[57] R. Jennings, "Cloud Computing with the Windows Azure Platform", Indianapolis, Wrox, 2009.

[58] National Institute of Standards and Technology (NIST), Secure Hash Standard, Federal Information Processing Standards publications 180-4, FIPS Pub.,United States, March, 2015.

[59] Rajeev Sobti and G.Geetha, "Cryptographic Hash Functions: A Review", IJCSI International Journal of Computer Science Issues, vol. 9, no. 2, 2012.

[60] Shamsiah binti Suhaili and Takahiro Watanabe, "Design of High-Throughput SHA-256 Hash Function based on FPGA", 6th International Conference on Electrical Engineering and Informatics (ICEEI), Langkawi, Malaysia, 2017.

[61] Kimmo Järvinen, Matti Tommiska and Jorma Skyttä, "Hardware Implementation Analysis of the MD5 Hash Algorithm", Proceedings of the 38th Hawaii International Conference on

System Sciences, IEEE, pp.1-10, 2005.

[62] M.Khalil , M.Nazrin and Y.W. Hau, "Implementation of SHA-2 Hash Function for a Digital Signature System-on-Chip in FPGA", International Conference on Electronic Design, Penang, Malaysia, December, 2008.

[63] Ricardo Chaves, Georgi Kuzmanov, Leonel Sousa and Stamatis Vassiliadis, "Improving SHA-2 Hardware Implementations", International Association for Cryptologic Research, Eds. Goubin and M. Matsui, pp. 298–310, 2006

[64] Edem Swathi, G. Vivek and G. Sandhya Rani, "Role of Hash Function in Cryptography", International Journal of Advanced Engineering Research and Science, pp. 10-13, 2016.

[65] Elena Andreeva and Bart Preneel, "A Three-Property-Secure Hash Function", International Workshop on Selected Areas in Cryptography, Selected Areas in Cryptography, Springer-Verlag Berlin Heidelberg, pp 228-244, 2009.

[66] Seong Oun Hwang, Intae Kim and Wai Kong Lee, "Hash Function", In book: Modern Cryptography with Proof Techniques and Applications, pp.87-102, 2021.

[67] Website: "Cryptography Hash functions", accessed 3 Oct, 2021,https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm.

[68] Justyna Brzezińska, "Singular Value Decomposition Approaches in A Correspondence Analysis with The Use of R", Folia Oeconomica Stetinensia, vol. 18, no. 2, pp. 178-189, 2018.

[69] Rowayda A. Sadek, "SVD Based Image Processing Applications: State of The Art, Contributions and Research Challenges" International Journal of Advanced Computer Science and Applications, vol. 3, no. 7, 2012.

[70] Abdulkadhem Abdulkareem Abdulkadhem and Tawfiq A. Al-Assadi, "Geo-Localization of Video Based on Proposed LBP-SVD Method", International Journal of Civil Engineering and Technology, vol. 10, no. 2, pp. 407-423, 2019.

[71] Abdulkadhem Abdulkareem Abdulkadhem & Tawfiq A. Al-Assadi, "Proposed a Content-Based Image Retrieval System Based on the Shape and Texture Features", International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 11, 2185- 2192, 2019.

[72] Ashish Kumar Bhandari, Anil Kumar and Prabin Kumar Padhy, "Enhancement of Low Contrast Satellite Images using Discrete Cosine Transform and Singular Value Decomposition", World Academy of Science, Engineering and Technology Journal, vol. 5, no. 7, pp. 707- 713, 2011.

[73] Meenakshi K, Ch. Srinivasa Rao and K. Satya Prasad, "A Fast and Robust Hybrid Watermarking Scheme Based on Schur and SVD Transform", International Journal of Research in Engineering and Technology, vol. 3, no. 4, 2014.

[74] Madhu B. and Ganga Holi, "An optimal and secure watermarking system using SWT-SVD and PSO", Indonesian Journal of Electrical Engineering and Computer Science, vol. 18, no. 2, pp. 917-926, 2020.

[75] Steven L. Brunton and J. Nathan Kutz, "Data driven science and engineering", University of Washington, US, 2017.

[76] Yuan Sun, Shiwei Ye, Yi Sun and Tsunehiko Kameda, "Exact and approximate Boolean matrix decomposition with column-use condition", International Journal of Data Science Analysis, vol. 1, pp. 199–214, 2016.

[77] Gregory D. and Pullman, N., "Semiring rank: Boolean rank and nonnegative rank factorizations", Journal of Combinatorics Information and System Sciences, vol. 8, pp. 223–233, 1983.

[78] Bˇelohlávek, R. and Trneˇcka, M., "From-below approximations in boolean matrix factorization: geometry and new algorithm", Journal of Computer and System Science, vol. 81, no. 8, pp. 1678–1697, 2015.

[79] Shireen Najeh Issa Odeh, "On Singular Value Decomposition of Rectangular Matrices", Master thesis, An-Najah National

University, Nablus, Palestine, 2009.

[80] H. R. Swathi, Shah Sohini, Surbhi and G Gopichand, "Image compression using singular value Decomposition", IOP Conference Series: Materials Science and Engineering, vol. 263, no. 4, 2017.

[81] Hiroaki Kotera, "RGB to spectral image conversion using spectral pallete and compression by svd", Proceedings 2003 International Conference on Image Processing, vol. 2, no. 3, pp. 461–464, Barcelona, Spain, 2003.

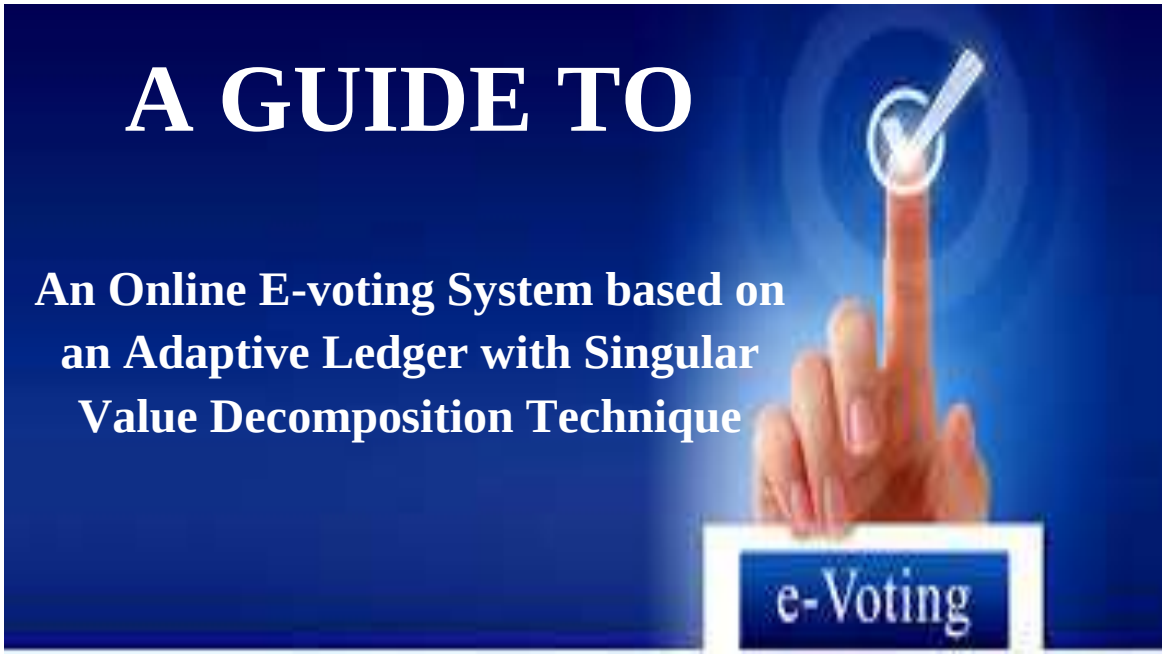
[82] Neethu.K. J. and Sherin Jabbar, "Using Approximate K-SVD Algorithm", International Conference on Innovations in Information, Embedded and Communication Systems, Coimbatore, India, 2015.

Appendix (A)

University of Babylon
College of Information Technology
Department of Software

A GUIDE TO

An Online E-voting System based on an Adaptive Ledger with Singular Value Decomposition Technique



Prepared by

Rihab Habeeb Sahib

Supervised by

Prof. Dr. Eman Salih Al-Shamery

2021

1. The Interfaces of the E-Voting System

The e-voting system consists of two main interfaces, the front-end interface which is a web application for the voter, and the back-end interface for the administrator as shown in the following sections:

1.1. The users' interface

The first interface that appears to the user (voter) is shown in figure (1), where the voter chooses to cast a vote or view the results if he/she voted previously.

Figure (1). The Front-end interface for voters to cast a vote or view the result

Figure (2) shows the next page where the voters enter their information, if an invalid ID is entered, a message box will appear

asking to enter a valid ID. Also, if the same ID is entered before, a message box will appear notifying the voter that he \she is allowed to vote only once.

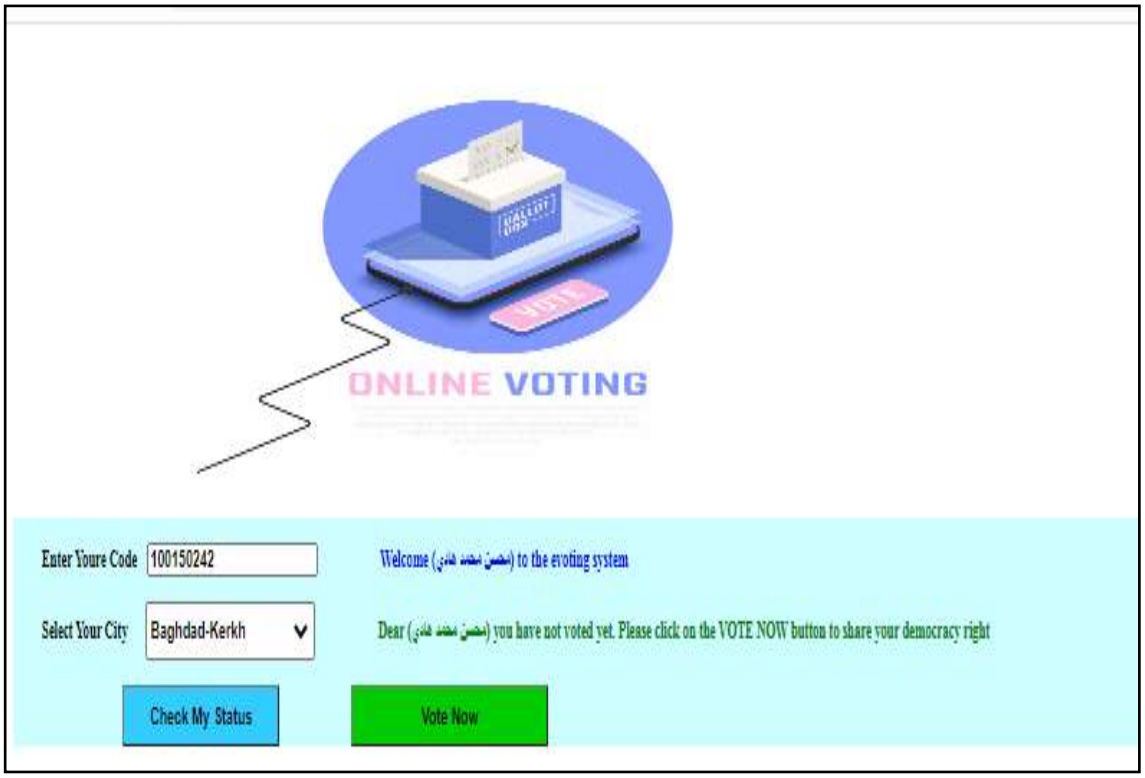


Figure (2). Information page for the voter to enter the system

When the voter enters a correct ID, a (Vote Now) button appears allowing to enter the voting page of 100 candidates as shown in figure (3). A voter clicks on the vote button corresponding to the desired candidate, a message box leading him/her to the (Drop vote in Ballot box) button. Figure (4) ensures the voter that he\she voted for candidate X and leading to the result page.

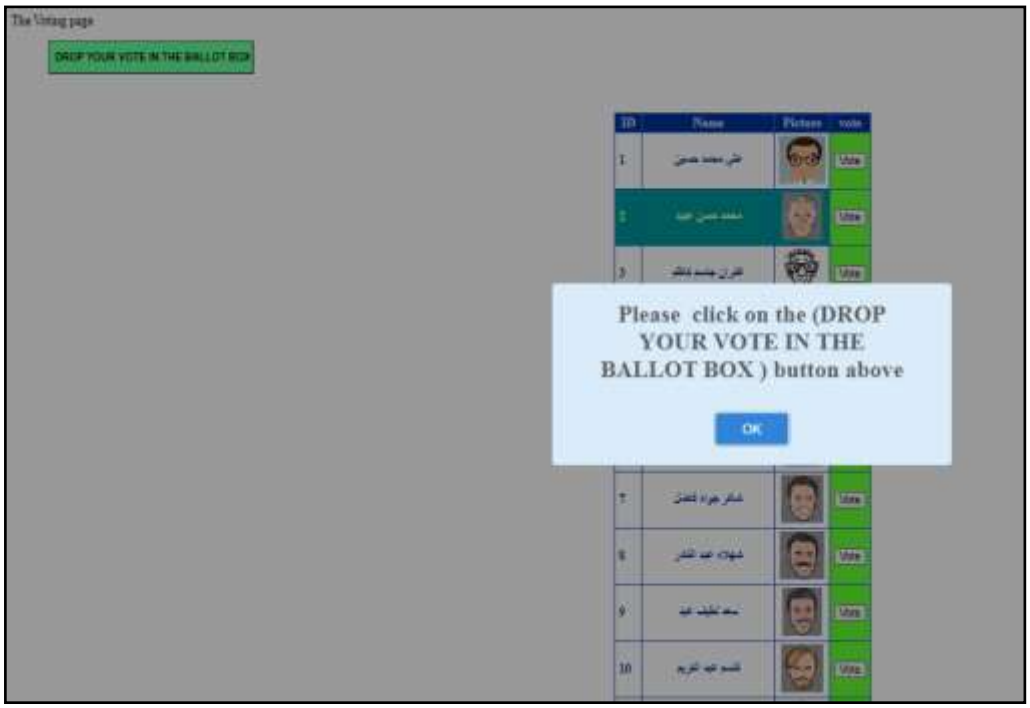


Figure (3). The voting page

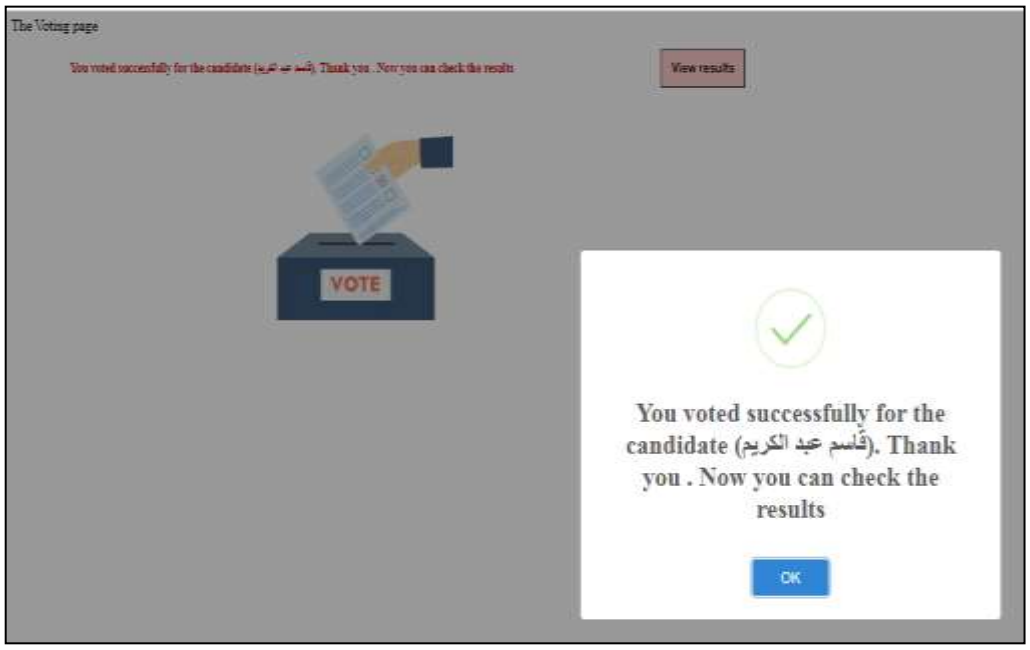


Figure (4). Ensuring the vote and leading to the result page

After voting, the voter will have the right to see the result as a table or a chart for transparency.

1.2. The Adminstrators' interface

The interface of the administrator is where surveillance occurs. The administrator has no control over the system as the system works dynamically in real-time. However, the admin has the ability only to set the number of transactions in a block. as shown in figure (5).

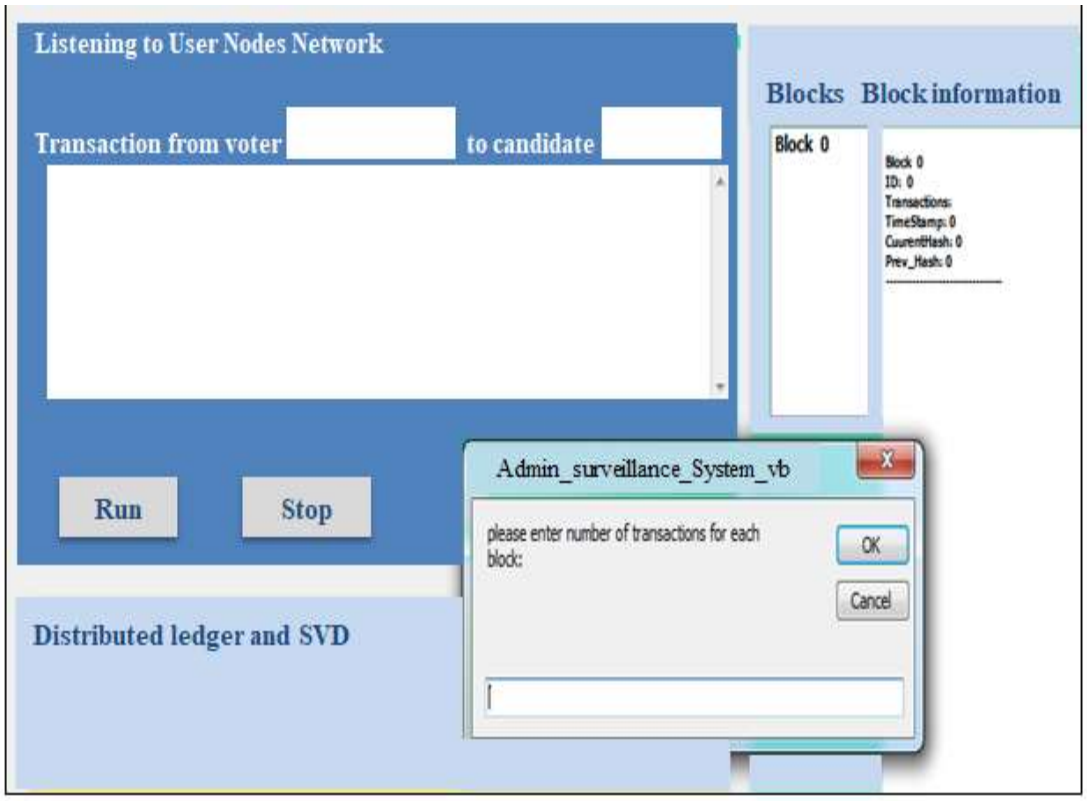


Figure (5). The Administrator interface

Note that the run and stop buttons are for testing the system before releasing it to work dynamically in real-time.

The administrator views how the e-voting process occurs by showing which ID voted to which ID of a candidate in the upper left window as a transaction and timestamp.

2. An example of an e-voting election event

The following table (1) and table (2) are part of the SQL database for the voters and candidates, respectively. For simplicity, the voter DB shows the first 30 voters, and the candidate DB shows the first 10 candidates.

No. of voter in SQL database	Voter Name	Voter ID
1	حسن علي جواد	100150162
2	حسين حمود علاوي	100150166
3	رجاء محمد حسن	100150170
4	تمار خالد حازم	100150174
5	مي عبد الرحيم	100150178
6	ورود فلاح خالد	100150182
7	دلال راجي حسين	100150186
8	فقار ثامر صالح	100150190
9	كريم ناجي يمان	100150194
10	شكرية عبد الاله	100150198
11	رونق صاحب علي	100150202
12	صابر عبد الرحمن	100150206
13	صبرية ابراهيم	100150210
14	ميادة عبد الجليل	100150214
15	صفا هاشم عبود	100150218
16	زينب غافل بسام	100150222
17	نور عبد الرضا	100150226
18	دعاء عبد الامير	100150230
19	حوراء مهدي	100150234
20	صالح ثامر عادل	100150238

21	محسن محمد هادي	100150242
22	هادي غفار	100150246
23	قاسم خالد	100150250
24	كرار محمد	100150254
25	علي جواد حسن	100150258
26	تمارة تركي	100150262
27	بلال جاسم	100150266
28	ثامر جواد علي	100150270
29	ضي عبد الرحمن	100150274
30	سجى قاسم	100150278

Table (1). The SQL DB for 30 voters

No. of Candidates in SQL database	Voter Name	Candidate ID	Counts	Photo of the candidate
1	علي محمد حسين	1	0	Images\can\p (1).png
2	محمد حسن عبيد	2	0	Images\can\p (2).png
3	غفران جاسم كاظم	3	0	Images\can\p (3).png
4	احمد رحيم كريم	4	0	Images\can\p (4).png
5	علي عبد الرحمن	5	0	Images\can\p (100).png
6	رغدة عبد علي	6	0	Images\can\p (5).png
7	شاكر جواد فاضل	7	0	Images\can\p (6).png
8	شهلاء عبد القادر	8	0	Images\can\p (7).png
9	سعد لطيف عبد	9	0	Images\can\p (8).png
10	قاسم عبد الكريم	10	0	Images\can\p (9).png

Table (2). The SQL DB for 10 candidates

Considering the following table (3) of votes, where every 10 transactions are grouped in a single block. In this example, voter 21 who's ID is 100150242 voted for candidate number 5: voter 5 who's ID is 100150178 voted for candidate 7, and so on.

	Serial No. of voters in the SQL database	Voter_ID	Candidate _ID
1	21	100150242	5
2	5	100150178	7
3	23	100150250	4
4	10	100150198	7
5	7	100150186	1
6	13	100150210	1
7	29	100150274	1
8	16	100150222	10
9	28	100150270	4
10	19	100150234	3
11	20	100150238	10
12	1	100150162	2
13	15	100150218	1
14	17	100150226	1
15	9	100150194	3
16	4	100150174	3
17	25	100150258	10
18	30	100150170	3
19	3	100150278	3
20	11	100150202	5
21	2	100150166	2
22	22	100150246	4
23	6	100150182	7
24	8	100150190	1
25	12	100150206	10
26	24	100150254	2
27	14	100150214	4
28	27	100150266	1

29	**18**	**100150230**	**10**
30	**26**	**100150262**	**10**

Table (3). Transactions of 30 voters (each color represents a block)

According to the 30 votes, figure (6) shows the results, in which candidate 1 has 7 votes; candidate 2 has 3 votes, and so on.

ID	Name	Picture	N_voters	Percentage
1	علي محمد حسين	Images\can\p (1).png	7	23
2	محمد حسن عبيد	Images\can\p (2).png	3	10
3	غفران جاسم كاظم	Images\can\p (3).png	5	16
4	احمد رحيم كريم	Images\can\p (4).png	4	13
5	علي عبد الرحمن	Images\can\p (100).png	2	6
6	رغدة عبد علي	Images\can\p (5).png	0	0
7	شاكر جواد فاضل	Images\can\p (6).png	3	10
8	شهلاء عبد القادر	Images\can\p (7).png	0	0
9	سعد لطيف عبد	Images\can\p (8).png	0	0
10	قاسم عبد الكريم	Images\can\p (9).png	6	20
11	رسول مجيد حميد	Images\can\p (10).png	0	0
12	نهى جبار ناهي	Images\can\p (11).png	0	0
13	غفار عبد الله محمد	Images\can\p (12).png	0	0
14	تقى هادي مهدي	Images\can\p (13).png	0	0
15	رنا عبد الإله	Images\can\p (14).png	0	0

Figure (6). The immediate results for 30 votes

When the administrator determines the number of transactions in a block the (run) button is clicked. The system will start listening to the voting processing fetching transactions as shown in figure (7).

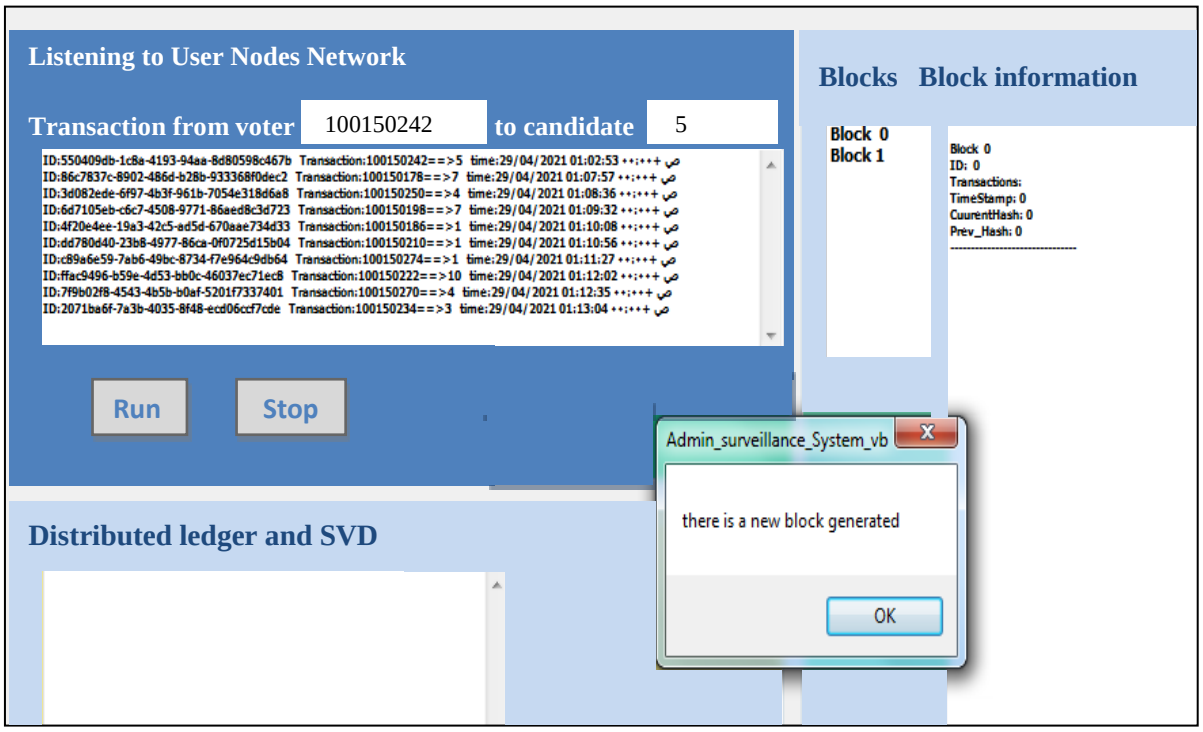


Figure (7). An interface showing the first 10 transactions

Each time a transaction arrives, a message box appears announcing the arrival of a new transaction, when the number of transactions reaches 10, a message box appears as shown above announcing that a new block is generated and name it Block1.

Each block consists of a block ID which is the addition operation of all voter IDs in the block, the transactions showing which ID voted to which Candidate number, a timestamp showing the date and the time of creating a block, a hash value using the hash function (SHA256) and the hash value of the previous block which is 0 for the initial block, as shown in figure(8).

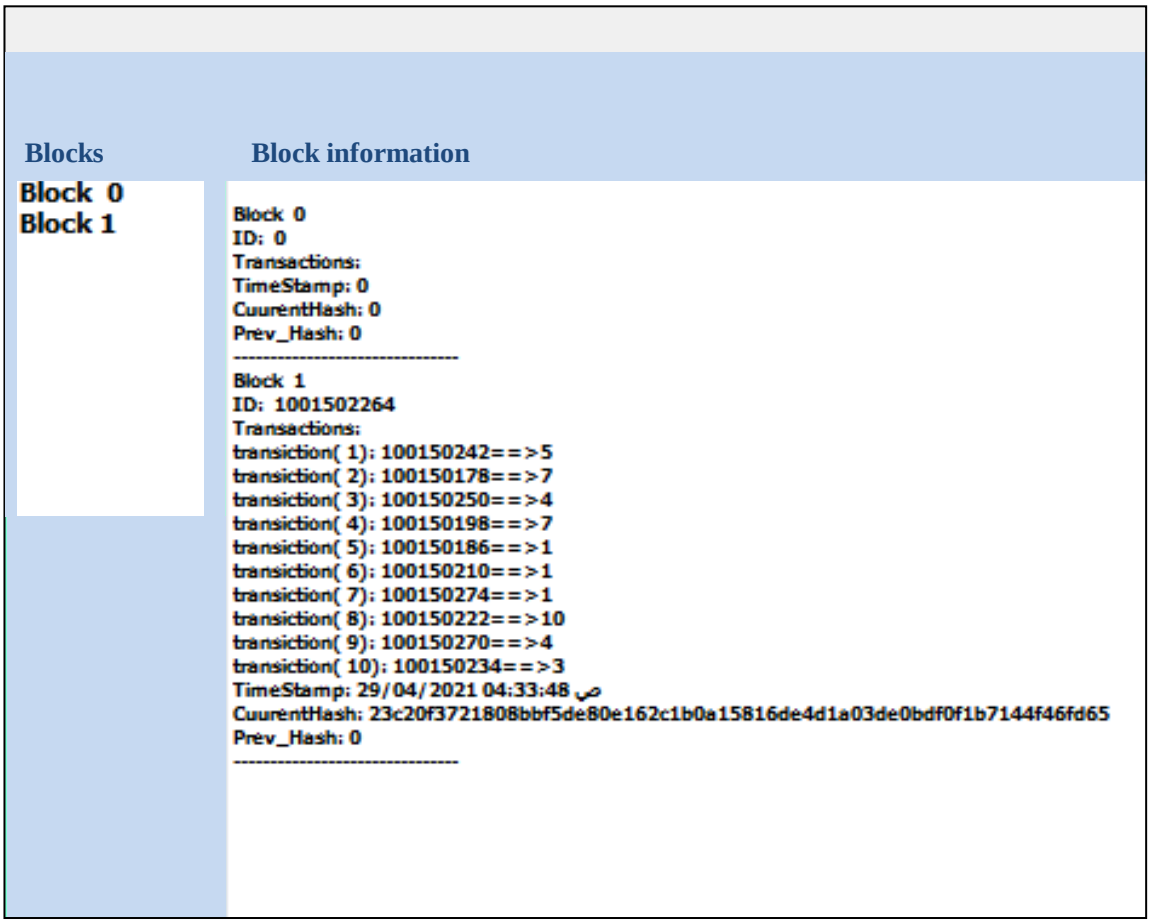


Figure (8). The first block of 10 transactions.

As soon as a block is created, another form of the results is created with the assistance of the singular value decomposition technique. The singular value decomposition has three outputs, the left matrix U stored in East Asia through the Azure Microsoft storage account, the Singular value matrix S stored in North Switzerland, and the right matrix V stored in Central Canada. Initially, these matrices contain zero values they are downloaded to the system, read, and written every time a new block is created. After reading the matrices, the ledger

is constructed by multiplying U*S*V' producing the ledger that is used later to match the results with the results in the SQL database.

SVD decomposition is applied to the updated ledger. (the ledger contain rows of candidates and columns of voters), each time a voter votes for a certain candidate the ledger is updated by replacing a 0 by 1 corresponding to the column of the voter with the row of the candidate. The U, S, and V matrices are written as files and cast back to the network nodes that are distributed in three different places as shown in figure (9). All these steps are repeated on every block till the end of the event.

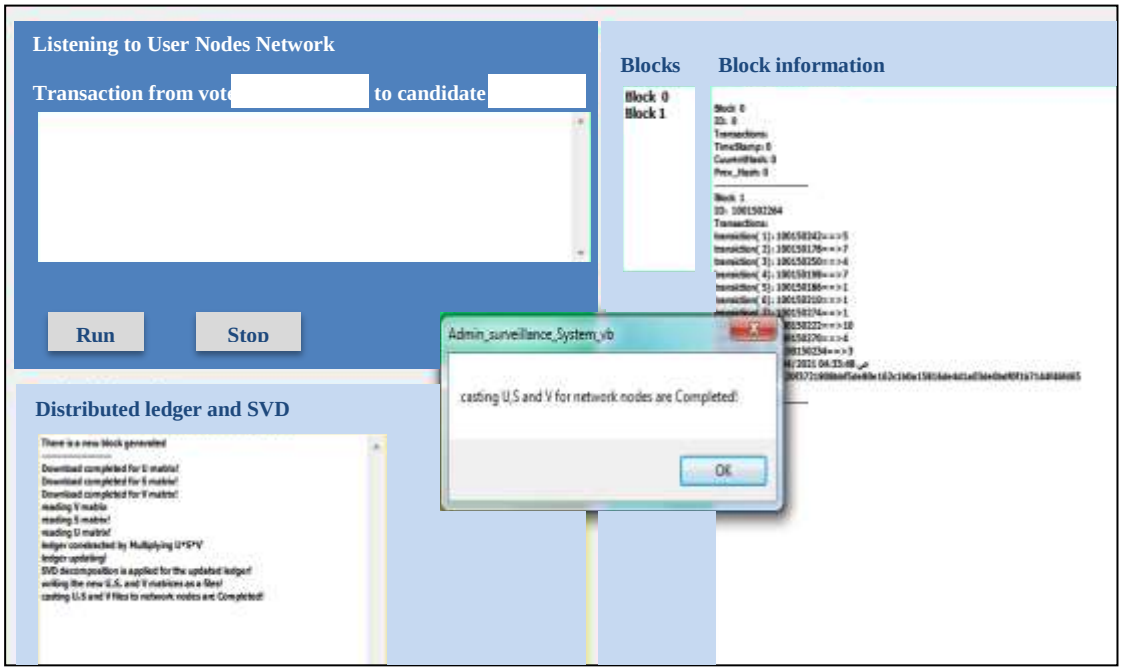


Figure (9). SVD was applied on the first block.

Each time a block is generated, a ledger is created by SVD and three matrices U, S, and V. In other words, Block1 will have (ledger1, U1, S1, and V1), Block2 will have the updated incremented (ledger2, U2, S2, and V2) and so on, as shown in figure (10).

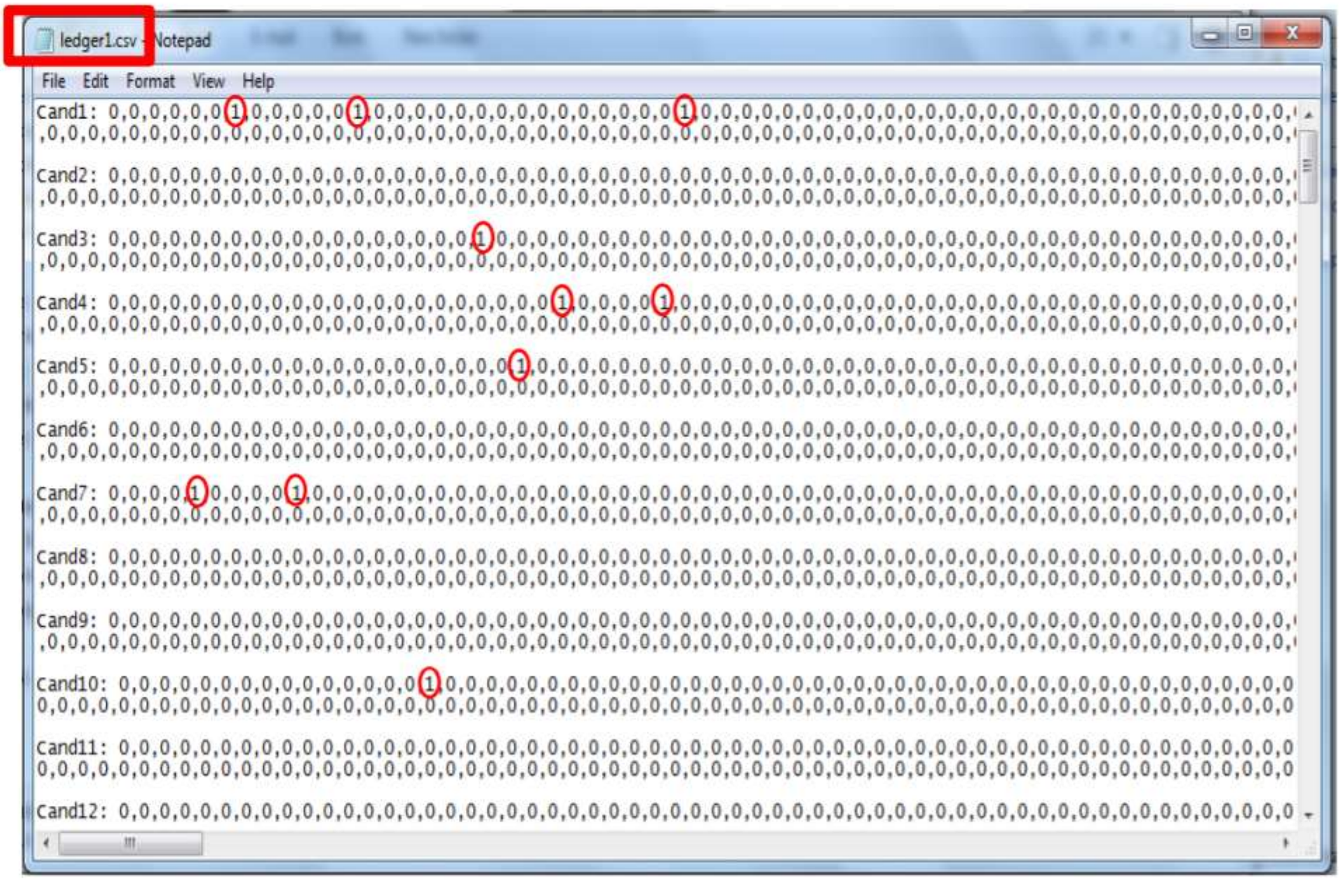


Figure (10). Ledger1for the first block

If we check the votes in block1 as shown in table (4), we find that the corresponding voter voted for the corresponding candidate, circled in red. For example, the voters in column7, 13, and 29 voted for candidate number 1, and so on.

	Serial No. of voters in the SQL database	Voter_ID	Candidate _ID
1	21	100150242	5
2	5	100150178	7
3	23	100150250	4
4	10	100150198	7
5	7	100150186	1
6	13	100150210	1
7	29	100150274	1
8	16	100150222	10
9	28	100150270	4
10	19	100150234	3

Table (4). Transactions of votes for block1.

Ledger 2 shown in figure (11) has the same scenario, updated and saving another incremented ledger with its' three matrices, U1, S1, and V1.

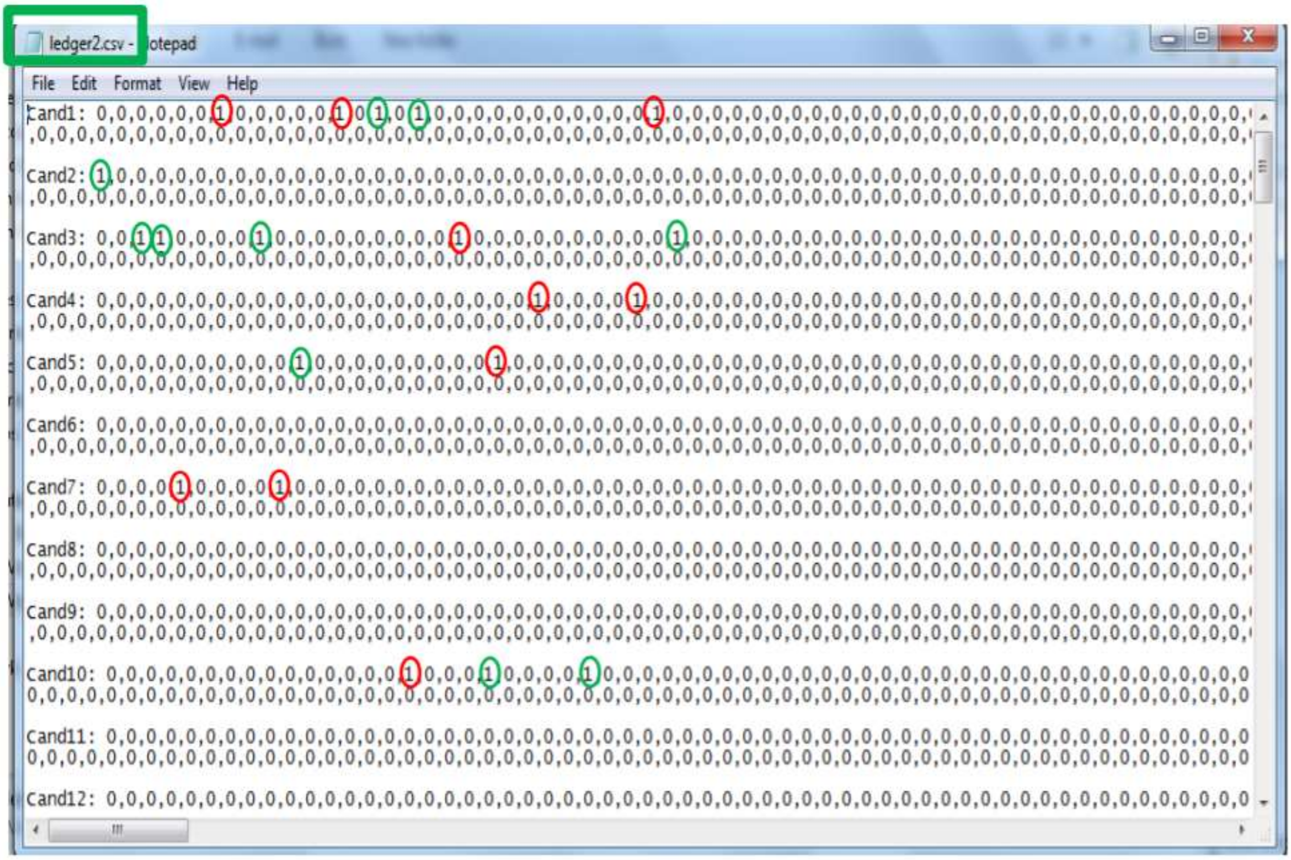


Figure (11). Ledger 2 for the second block.

In ledger 2, for example, we see that voters 15 and 17 voted for candidate 1, circled in green, voter 1 voted for candidate 2, based on the transaction of votes in table (5), and so on. Again this ledger will have a copy of U2, S2 and V2.

	Serial No. of voters in the SQL database	Voter_ID	Candidate _ID
11	20	100150238	10
12	1	100150162	2
13	15	100150218	1
14	17	100150226	1

15	**9**	**100150194**	**3**
16	**4**	**100150174**	**3**
17	**25**	**100150258**	**10**
18	**30**	**100150170**	**3**
19	**3**	**100150278**	**3**
20	**11**	**100150202**	**5**

Table (5). Transaction of votes for the second block.

The final ledger is ledger 3 as shown in figure (12), in which voter 2 voted for candidate 2, voters 8 and 27 voted for candidate 1circled in blue, and so on. Based on the transaction of votes for the third block shown in table (6).

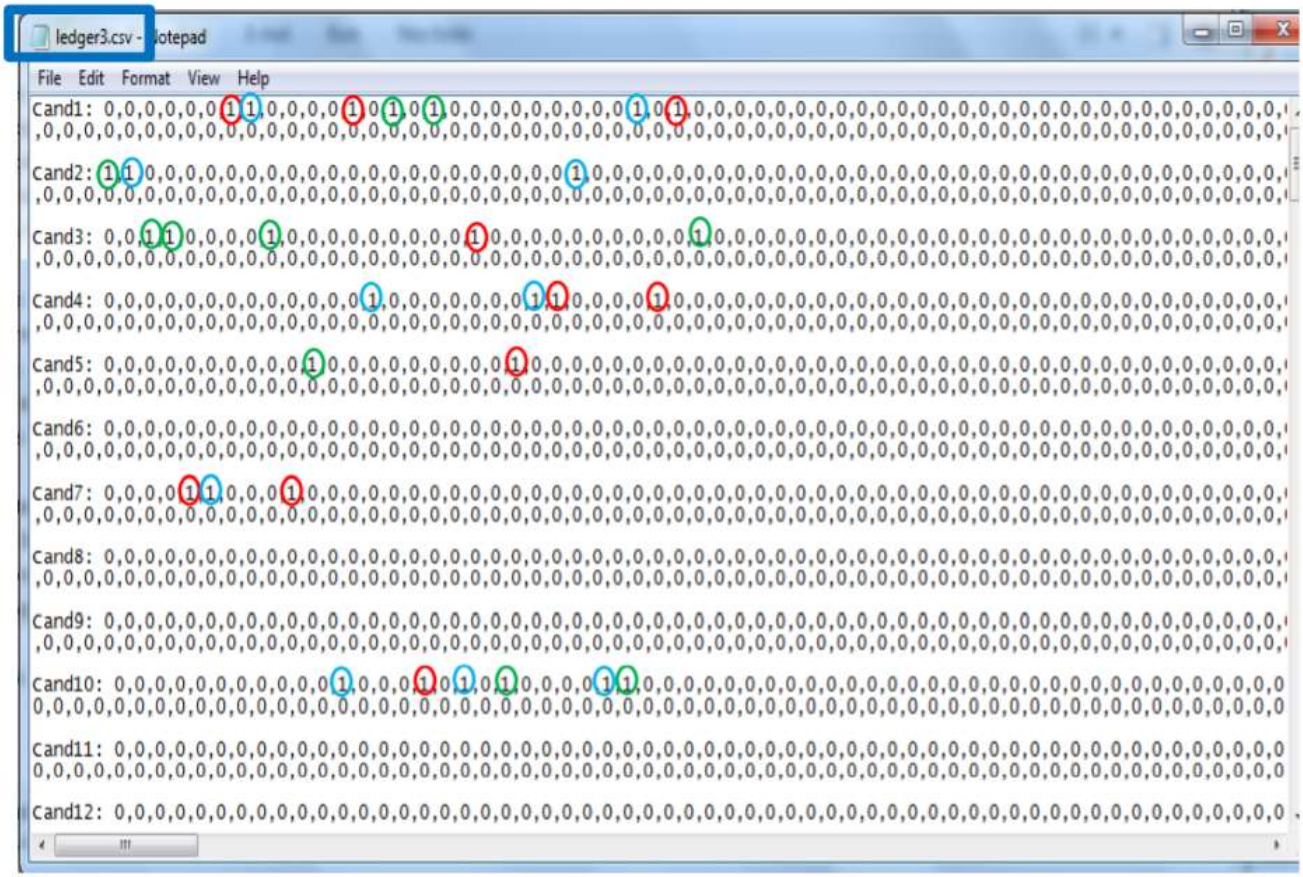


Figure (12). Ledger 3 for the third block.

	Serial No. of voters in the SQL database	Voter_ID	Candidate _ID
21	2	100150166	2
22	22	100150246	4
23	6	100150182	7
24	8	100150190	1
25	12	100150206	10
26	24	100150254	2
27	14	100150214	4
28	27	100150266	1
29	18	100150230	10
30	26	100150262	10

Table (6). Transaction of votes for the third block.

The values get more complicated as long as the votes increase because, in SVD, each value in each row is related to all values in each row, resulting in real complicated coefficients.

Our SVD is based on rank r that showed a decomposed matrix with no loss of results. According to the rank =r for 100 candidates and 100000 voters, instead of downloading reading, and writing a matrix with size 10000000, we deal with size U*r + r*r + V*r.

All matrices are shown as the last update for that last ledger, U3, S3, and V3 are shown in figure (13).

Matrix U3 in East Asia

	A
1	0
2	0
3	0
4	-1
5	0
6	0
7	0
8	0
9	0
10	0
11	-1
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0

S3 Matrix in North Switzerland

	A
1	2.236068
2	0
3	0
4	0
5	2.236068
6	0
7	0
8	0
9	1.732051
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	

V3 Matrix in Central Canada

	A
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	-0.44721
9	0
10	0
11	0
12	0
13	0
14	-0.44721
15	0
16	-0.44721
17	0
18	-0.44721
19	0
20	0
21	0
22	0
23	0
24	0

Figure (13). U3, S3, and V3 for ledger3 of the third block.

Appendix

(B)

The online results of the election, matching the counts in the final ledger.

ID	Candidate Name	N. of votes	Percentage
1	علي محمد حسين	1	6
2	محمد حسن عبيد	2	13
3	غفران جاسم كاظم	0	0
4	احمد رحيم كريم	0	0
5	علي عبد الرحمن	2	13
6	رغدة عبد علي	0	0
7	شاكر جواد فاضل	0	0
8	شهلاء عبد القادر	1	6
9	سعد لطيف عبد	0	0
10	قاسم عبد الكريم	0	0
11	رسول مجيد حميد	0	0
12	نهى جبار ناهي	0	0
13	غفار عبد الله محمد	0	0
14	تقى هادي مهدي	0	0
15	رنا عبد الاله	0	0
16	ضمياء حسن علي	0	0
17	منى عبد الرزاق	0	0
18	لمياء قصي باقر	0	0
19	حميدة عبد الامير	0	0
20	بسام فاضل عباس	0	0
21	عباس عبد القاسم	0	0
22	ناجي هاني مسعود	0	0
23	مصطفى مهدي محمود	0	0
24	حسن حسين ستار	0	0
25	نادية حسون عامر	0	0
26	ضحى جواد محسن	0	0
27	كرار غازي كريم	0	0
28	نجم سلوان طاهر	0	0

29	يعقوب تحسين غالي	0	0
30	سناء خالد رحيم	0	0
31	يوسف ثائر حمود	0	0
32	هناء فارس ثامر	0	0
33	صفاء طه عبد الزهرة	0	0
34	شيماء وائل	0	0
35	ظافر عبد الواحد	0	0
36	نور عبد القاسم	0	0
37	عبد الصمد راجي	0	0
38	سيف باقر علي	0	0
39	فاطمة عبد الزهرة	0	0
40	كاظم عبد الكريم	1	6
41	غالي جاسم جواد	0	0
42	فاطمة تمار ميثم	0	0
43	زينب قادر حسني	0	0
44	نبراس بلال	0	0
45	باسم عبد الواحد	0	0
46	راجي غازي علوان	0	0
47	محسن صفاء محمد	0	0
48	نضال عبد الغفار	0	0
49	حسام حسين ثائر	0	0
50	جليل عبد علي	0	0
51	صفا نزار عبد علي محمد	0	0
52	ضحى فهمي علي	0	0
53	ضحى مهدي حلبوص عبد	0	0
54	ضياء فارس مسلم عبادة	0	0
55	طيبة سليم دعبول عبيد	2	13
56	عذراء ستار عباس كاظم	0	0
57	علا عبد الكريم حسين جبار	0	0
58	علي ثابت وحيد عجيل	0	0
59	علي جعفر كاظم عباس	0	0

60	علي حسين عبد السادة خطار	1	6
61	علي حسين هادي بشن	0	0
62	علي ظاهر صلال عبود	1	6
63	علي عباس رزوقي عطيوي	0	0
64	علي فائز حسوني عبيد	0	0
65	علي ناظم حسين	0	0
66	علي ماجد عبد الحسين أيوب	0	0
67	علياء ميثم حاتم	0	0
68	عمر وهام مناحي صياح	0	0
69	فاطمة الزهراء حيدر سليم قنديل	0	0
70	فاطمة عماد حميد حمد	0	0
71	فاطمة لطيف سلمان حسون	0	0
72	فرح فاضل هاشم فزع	0	0
73	فريال ماجد محمد أمين	0	0
74	قاسم شارع مياح شطب	0	0
75	قائد كريم رحمن نجم	0	0
76	ماذن محمد عبيس بعيو	0	0
77	محمد علي حسين ذياب حسين	1	6
78	مخلد عبد الحسن كافي حسان	0	0
79	منتظر داخل ناصر حسين	0	0
80	ميلاد حسين هلال	0	0
81	نبراس عماد عطيوي حمد	0	0
82	نجوان غسان عدنان جابر	0	0
83	ندى كاظم جبر معيدي	0	0
84	نغم عبد الله جرد	0	0
85	نوال محمد جابر راضي	0	0
86	نور حسن زاجي نجم	0	0
87	نور صاحب عايز	0	0
88	نور طالب عبد عبد الله	0	0
89	نور علي حمزة	0	0
90	نور فاضل علوان عواد	0	0

91	نور معن هادي محمد	0	0
92	هاجر عمران حمزة عمران	0	0
93	هبة أحمد فالح سركال	0	0
94	هدى صادق رسول	0	0
95	هبة صبحي صبار كاظم	0	0
96	هدير فلاح حسن ديكان	0	0
97	هديل علاء سعيد كشاش	0	0
98	هند حيدر حميد كبيحان	0	0
99	هيام عبار كردوش عضد	0	0
100	هيثم عدنان حميد عبد	3	20

الخلاصة

تعتبر الانتخابات حدثًا مهمًا يحدث في العديد من البلدان. فالانتخابات الورقية مضيعة للوقت في فرز الأصوات وتكلفة المواد وغير موثوق بها ولا يمكن ضمان الخصوصية حيث يمكن التلاعب بالأصوات نظرًا لقدرة لجنة الانتخابات التحكم في العملية. قد تحسب أنظمة التصويت الإلكترونية النتائج في وقت أقل ، وتكلفة أقل للمواد ، وقد تحافظ على خصوصية المواطنين ، لكنها لا تزال تعتبر غير موثوقة حيث يمكن التلاعب بالأصوات.

يتعامل النظام المقترح مع سجل ذي الحجم الضخم باستخدام تقنية تفريق القيم الفردية (SVD) كأداة لتقليل البيانات والأبعاد، وأداة لقياس شفافية النظام من ناحية مطابقة نتائج الانتخابات مع السجل الذي يتم إنشاؤه بمجرد تكوين بلوك من الأصوات يحمل النتائج الفورية. أيضًا، يتم استخدام تقنية SVD كأداة لتوزيع السجل عن طريق إرسال مخرجاته إلى عقد شبكة منفصلة. في كل مرة يتم فيها إنشاء بلوك من الأصوات، يتم أيضًا إنشاء سجل لذلك البلوك ويحمل نسخة من النتائج في نموذج آخر عن طريق تطبيق عمليات تقنية SVD.

يتكون النظام من منصات مستخدم ومسؤول موزعة على أربع مراحل. المرحلة الأولى هي مرحلة ما قبل المعالجة والتي تشمل إعداد قوائم المرشحين والناخبين ، وبناء العقد الشبكية الموزعة لتخزين ومعالجة البيانات ، وإنشاء السجل العام الذي يحتفظ بالنتائج التي يتم توزيعها باستخدام الخدمات السحابية. المرحلة الثانية هي مرحلة التأكيد حيث يُسمح لكل مواطن بالتصويت بناءً على بطاقة الهوية الدولية والإدلاء بصوته مرة واحدة فقط.

المرحلة الثالثة هي مرحلة التصويت الإلكتروني التي تشمل المراحل الرئيسية للنظام المقترح ، وتشمل إضافة المعاملات (الأصوات) إلى البلوك، واسترجاع البيانات من عقد الشبكة الموزعة للحصول على والتحقق من صحة وتحديث السجل باستخدام SVD.

المرحلة النهائية هي مرحلة النتيجة ، والتي تُظهر نتائج الانتخابات كجزء من الشفافية للناخبين بنجاح ومطابقتها مع السجل النهائي الذي يحمل النتائج التي شكلها SVD.

يتم تطبيق النظام على 100000 ناخب و 100 مرشح وكل بلوك احتوى على 20-100 معاملة (أصوات). صوّت مائة متطوع لاختبار النظام والتحقق من نتائج ناخبين مختلفين ومرشحين مختلفين بشكل عشوائي.

جمهورية العراق
وزارة التعليم العالي و البحث العلمي
جامعة بابل/ كلية تكنولوجيا المعلومات

تطوير السجل الموزع القابل للتكيف المعتمد على تفريق القيم الفردية لنظام تصويت الكتروني

رسالة

الى مجلس كلية تكنولوجيا المعلومات / جامعة بابل وهي جزء من متطلبات نيل شهادة الماجستير في تكنولوجيا المعلومات/ برمجيات

من قبل

رحاب حبيب صاحب نهر

بإشراف

أ.د. ايمان صالح الشمري

1442هـ 2021 م