Blockchain based Electronic Voting System

Project report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Moulik Chaturvedi (201326) & Garvita Sharma (201122)

Under the supervision of

Dr. Aman Sharma

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

DECLARATION CERTIFICATE

I hereby declare that the work presented in this report entitled **'Blockchain based Electronic Voting System'** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to December 2023 under the supervision of **Dr. Aman Sharma** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Moulik Chaturvedi (201326) Garvita Sharma (201122)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Aman Sharma,
Assistant Professor (SG),
Department of Computer Science & Engineering and Information Technology
Dated:

ACKNOWLEDGEMENT

First, We express my heartiest thanks and gratefulness to God for His divine blessing to make it possible to complete the project work successfully.

We are grateful and wish our profound indebtedness to Dr Aman Sharma, Assistant Professor (SG), Department of CSE & IT, Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of our supervisor in the field of "Cyber Security and Cloud Computing" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We are deeply indebted to Professor Vivek Kumar Sehgal, Head, Department of CSE & IT at the Jaypee University of Information Technology for his constant encouragement and motivation that enthused me to complete our project with zest and determination. We would like to express our sincere appreciation to Dr Aman Sharma, Assistant Professor (SG) in the Department of CSE and IT at the Jaypee University of Information Technology, for their insightful recommendations during the course of our project.

We would also generously acknowledge each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.
No expression of appreciation is complete without recognition of the prayers, good wishes, advice and moral support of our affectionate parents which helped us immensely to achieve our goal.

Name: Moulik Chaturvedi
Roll No: 201326

Name: Garvita Sharma
Roll No: 201122

Project Group No. 12

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1. VVPAT - Voter Verifiable Paper Audit Trail
2. EVM - Electronic Voting Machine
3. E-Voting - Electronic Voting
4. AES - Advanced Encryption Standard
5. SHA - Secured Hashing Algorithm
6. ECC - Elliptical Curve Cryptography
7. RSA - Rivest, Shamir, Adleman
8. CRHF - Collision Resistant Hash Functions
9. OWHF - One Way Hash Functions
10. TWHF - Two Way Hash Functions
11. VoterID - Voter Identification

ABSTRACT

The idea of blockchain makes the eVoting system more secure and reliable. This technology, for its security and reliability has always been a go to method for solving the problem of unfair voting and elections. Although we have some really good research and implementations of the technology, they still have some basic problems which we have addressed and tried to resolve in this paper. The lack of voter anonymity in eVoting systems makes a loophole that can be exploited through ways outside the system. In contrast, some digital voting technologies are significantly more efficient than the current blockchain-based score voting systems. Some further problems with the current options are that they are not suitable for immediate deployment, are hard to use, and have inflexible or nonexistent tallying processes. We have used, up to date hashing and encryption techniques to preserve the voter data, even while keeping the data publicly available to make the system both secured and reliable. Our proposed model ensures the inexpensive computation and provides a flexible tallying mechanism.

Chapter 1: INTRODUCTION

1.1 INTRODUCTION

In the 21st century, Electronic voting has been a field that has been researched for several years. Election administration could undergo a radical change thanks to electronic voting, or "e-voting" technologies. As can be observed, E-Voting is already being accepted as the way to go. Countries worldwide have switched to e-voting solutions in one way or the other. India is one such example. By implementing the Electronic Voting Machines, commonly known as EVMs, India introduced new standards for a faster and more secure voting mechanism for the country and even though there are allegations made quite often, it has been found that the majority of them are baseless and the rest can not be proved. On the other hand, the conventional voting methods have long been beset by issues including fraud, inefficiency, and complicated logistics.

When we discuss the inefficiencies of the traditional voting mechanisms, we see problems like vote manipulation, slow implementation of the process of casting votes, and even slower implementation of tallying of votes. Vote manipulation raises concerns about the integrity of the elections, and since there is often one major entity that governs the conducting of the elections it is easy to believe that the election can be rigged and the results will be biased in favour of the will of the governing entity. The other problem, as mentioned above, is the slow implementation of the whole process.

To address the slow implementation of the process of voting, let us see a short case study on the election process that was used in India before the introduction of EVMs and the process that is used today, with EVMs. Before the introduction of EVMs, the concept of paper ballot was used. A voting (paper) ballot by the definition is a confidential piece of paper on which the voter writes its vote and then the paper is secretly put in one of the sealed boxes in a polling booth. These boxes are then taken into possession by officials and are heavily guarded so that there are no chances of breach in security. The votes are then taken to a counting station which is also heavily guarded, where the sealed boxes are opened and the votes are counted manually, one by one. Even though the security of the

process is maintained, the corruption amongst the officials and booth capturing cannot be ignored. This made the formerly used manual voting system by paper ballot questionable.

To counter this and to provide a more secure and a faster voting mechanism, The government of India brought a new system which leveraged the power of Electronic Voting Machines. These machines were developed and produced by the government owned companies, Electronics Corporation of India and Bharat Electronics. These machines introduced high security features like “limiting the rate of casting votes to five per minute”, “security lock-close”, and an electronically maintained database of “voting signatures and thumb impressions”. After the initial introduction of EVM, which included only an Electronic Voting Machine ballot Unit, and a respective Control Unit, various parties that lost the elections alleged that the machines are faulty or rigged so the votes are not going to the candidate the voter actually voted for. To solve this issue, the Election Commission developed EVMs with voter-verified paper audit trail (VVPAT) systems. This system helped the voter to verify that the vote has been given to the candidate they actually pressed the button for.

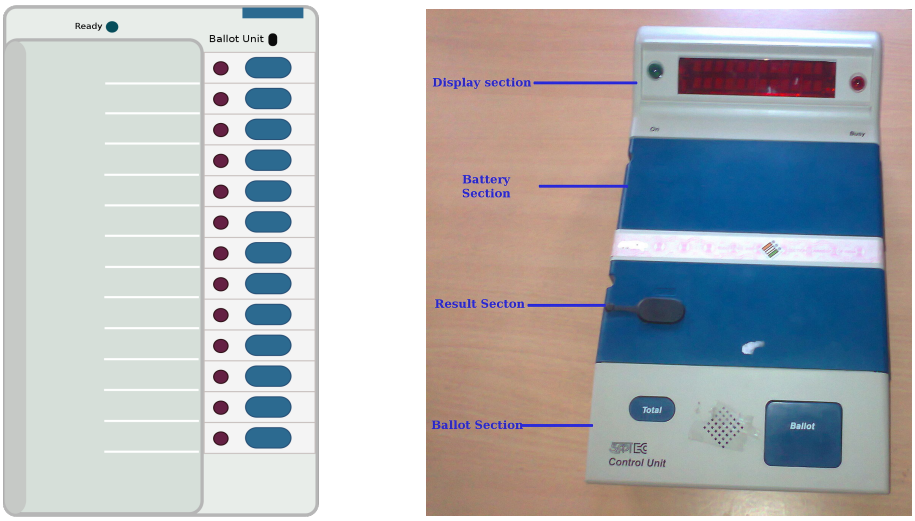

Figure 1.1 EVM Machine(left) and Control Unit(right)

(Source: Wikipedia)

Even though the Election Commission of India states that their machines and election protocols are tamper-proof, a number of allegations are still made from time to time and

because these allegations are made by political parties and leaders with a huge following and influence on millions of citizens, it creates an aura that the elections can be rigged and it is a logical statement to make that if the voters do not believe in the integrity and efficiency of an election process, it is in a way, unnecessary to have the elections at all.

We researched and found that this integrity can be maintained by using the Blockchain technology. The problems of mutable votes, booth capturing, election integrity, corruption and unauthorised votes can be handled easily with a well designed Blockchain based electronic voting system. Also, apart from solving the above stated problems, a blockchain based e-voting system will help in an overall faster, easier to use, and more efficient way of conducting voting at any level worldwide.

It is not the first time that such a project has been undertaken, in fact the concept of blockchain based electronic voting systems is now used for elections on a wide scale in some societies. Sierra Leone became the world's first country to conduct a Blockchain-based voting on March 7, 2018. Russia too, launched a pilot project on blockchain-based electronic voting system in June, 2019.

With all these advances in the field of blockchain, and keeping in mind the need of the hour, it is evident that we need a system or a framework that can ensure efficient voting while taking in account all the security measure necessary. Hence we, bring forward this project presenting a "Blockchain based e-Voting System Framework".

1.2 PROBLEM STATEMENT

A variety of democratic institutions use different kinds of voting methods, but they are beset by problems including security flaws, transparency, and practical difficulties. The demand for a contemporary, dependable electronic voting (eVoting) system that allays these worries and preserves the integrity of the democratic process is growing as technology develops.

After noticing the issue, we set out to create a transparent and safe blockchain-based electronic voting system. By using blockchain technology, data transparency, immutability, and resistance to tampering are ensured. The blockchain would record each vote as a transaction, encrypt it, and store it securely, making it very difficult for nefarious actors to tamper with the results. It would be possible for voters to independently confirm the correctness of their ballots cast on a public, secure blockchain. Public keys would enable transparent validation, but each voter would have a distinct private key for encryption and authentication.

Election procedures could be completely changed by creating a transparent and safe blockchain-based electronic voting system. Through the resolution of security flaws, improved openness, and guaranteed accessibility, this approach has the potential to rebuild public confidence in the democratic process. A multidisciplinary team of specialists in blockchain technology, encryption, and election procedures must work together to create and build a strong solution that protects election integrity and offers all citizens a smooth and user-friendly voting experience.

1.3 OBJECTIVES

1. Design and implement a blockchain-based e-Voting system that guarantees secure and transparent vote recording and tallying.
2. Provide a user-friendly and accessible interface for voters to cast their votes while ensuring their anonymity.
3. Establish a robust consensus mechanism to validate and record transactions securely on the blockchain which also allows the authorities to verify the accuracy.

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

As we have discussed in the previous sections, the growing technology and needs of democracy demands an e-Voting system that is secured and efficient in its woking. The significance of this project can be demonstrated by addressing points like electoral integrity, building trust in the democratic process, mitigation of security risks, accessibility and inclusivity, and technological advancements in governance.

Using a voting system based on blockchain technology is essential to solving the ongoing problems with electoral integrity. The project aims to provide a safe and transparent platform that can drastically lower the risks connected with fraud and manipulation in conventional voting systems by utilising the decentralised and tamper-resistant features of blockchain. The foundation of any democratic society is trust. The adoption of a voting system based on blockchain technology fosters trust in the election process by offering an unchangeable and transparent voting record. Election officials, candidates, and voters all benefit from this transparency, which strengthens the validity of democratic results.

The initiative tackles the risks associated with hacking and illegal access, which are intrinsic weaknesses of traditional voting systems. A blockchain-based voting system guarantees a high level of security by utilising cutting-edge cryptographic techniques and decentralisation, protecting the integrity of the entire electoral infrastructure. The project adds to the current revolution in governance brought about by technology. The voting system's adoption of blockchain technology not only modernises election procedures but also establishes a standard for the incorporation of creative solutions to deal with social as well as security issues. The initiative is now at the forefront of the democratic practices' progress thanks to this advancement.

Lastly, The voting process can become more inclusive and accessible by utilising blockchain technology. The project intends to investigate the possibility of secure and remote voting, enabling people who are physically or geographically unable to vote to take part in the democratic process. The values of civic engagement and equal representation are in line with this inclusivity.

Looking at the significance of this project and the scale on which the democracies worldwide, are trying to implement such a framework, we were motivated to provide a viable solution. This project is driven by a thorough comprehension of the structural issues that plague traditional voting systems. The idea that technology can spur positive change serves as the foundation for the motivation.

Through the use of blockchain technology, the project aims to revolutionise voting systems by providing a solution that meets the needs of the digital age and encourages innovation in the field of governance. The project's goal is to create a blockchain-based voting system that is both scalable and flexible, with the potential to serve as a model for other nations looking to improve the security and transparency of their election systems.

1.5 ORGANISATION OF PROJECT REPORT

The report is organised as follows:

- Chapter-02 outlines the existing related work in the field of Blockchain Technology and E-Voting Systems. It further presents the outputs which we eventually compare and discuss in this report.
- Chapter-03 puts forward the system that is formulated to solve the problem statement regarding this project i.e. a Blockchain based eVoting System and is designed to work so as to improve security. This is where we cover the software requirement and security factors, whilst also addressing the efficiency and usability of our system.
- Chapter-04 puts forward the analysis of the results in depth and also with content to existing work in the field.
- Finally, Chapter-05 presents the conclusion of the study. It also contains the application contribution with future scope.

Chapter 2: LITERATURE SURVEY

In this section we have covered various studies related to the latest developments in the field of Blockchain based eVoting Systems [4,5,6].

2.1 OVERVIEW OF RELEVANT LITERATURE

Numerous recent studies have looked into the possibility of using blockchain[10] technology for electronic voting. For example, a 2023 World Bank research found that blockchain[10] technology might help improve the security, efficiency, and transparency of elections in developing countries. The study also found that electronic voting systems based on blockchain technology[4,5,6] might help reduce election fraud and increase voter turnout.

A blockchain-based electronic voting system that ensures the security and integrity of the voting process while protecting voters' privacy was developed and tested in a separate recent study.. This study was published in the journal Nature [12] in 2023. According to the study, the suggested system may survive a range of assaults, such as man-in-the-middle and denial-of-service attacks.

Before blockchain-based voting systems are widely used, a few issues[5] still need to be resolved, despite their potential advantages. One issue is that, in the context of electronic voting, blockchain technology is still somewhat new and unproven. Another difficulty is that creating and implementing an electronic voting system based on blockchain technology might be costly and complicated. The potential advantages of blockchain-based electronic voting systems outweigh these difficulties. We may anticipate seeing an increasing number of nations and organisations use blockchain-based electronic voting systems as the technology advances and matures in order to increase the efficiency, security, and transparency of their elections.

This investigation benefits greatly from the "ACB vote” [11] research paper, which provides information about the state of blockchain-based electronic voting systems as of

right now. Our study intends to expand on the groundwork established by this reference by delving deeper into the subtleties of blockchain technology, addressing particular issues and offering practical suggestions for the broad implementation of safe and effective blockchain-based electronic voting systems. The important conclusions of the cited studies will be covered in more detail and placed in the larger context of blockchain technology and electronic voting in the sections that follow.

While examining the terrain of blockchain-based electronic voting systems, it is critical to recognise the complex issues that need to be resolved in addition to the exciting prospects. The potential susceptibility of blockchain systems to new cyberthreats and attacks is an important factor to take into account. The Nature study [12] emphasises the robustness of a well-thought-out blockchain-based voting system, but continued advancements in cybersecurity are necessary to keep up with ever-more-advanced threats. Blockchain developers and cybersecurity specialists must work together to strengthen the technology against new threats and guarantee blockchain's long-term viability in preserving the integrity of electoral processes.

In addition, scalability becomes a significant concern when it comes to national elections with sizeable voter bases. For e-voting systems to be implemented practically, blockchain networks must be able to manage a sizeable amount of transactions[5,6], especially during periods of high voting volume. Solutions for blockchain scalability innovations like sharding and layer-two protocols offer ways to get around these problems. Resolving scalability issues is essential to achieving national adoption of blockchain-based electronic voting systems and building trust in the technology's capacity to manage the pressures of large-scale, real-world elections.

A thorough analysis of the socioeconomic effects of switching to blockchain-based electronic voting systems is also necessary. Blockchain has the potential to make voting more democratic, but there are still worries about the digital divide. To avoid denying certain demographic groups the right to vote, it is imperative to guarantee equitable access to technology and digital literacy. To foster an inclusive atmosphere that will support the

growth of blockchain-based electronic voting, governments and organisations must take proactive measures to address these disparities through infrastructure development and educational programs.

Looking at successful case studies and worldwide trends in addition to individual research offers a more comprehensive understanding of the adoption trajectory of blockchain-based electronic voting. Countries like Estonia [13] have led the way in incorporating blockchain technology into their election procedures, demonstrating the usefulness and efficiency of this technology. Estonia's experience shows that blockchain [13] can be seamlessly integrated into current infrastructures, providing a transparent and safe platform for electronic voting, with careful planning and a commitment to cybersecurity.

The World Bank's acknowledgement of blockchain technology's potential in developing countries further emphasises the technology's worldwide significance. Blockchain appears as a tool that can empower citizens and fortify democratic institutions as developing nations confront particular difficulties in guaranteeing free and transparent elections. Case studies of countries that have successfully adopted blockchain-based electronic voting systems provide useful standards, providing information on best practices and possible drawbacks.

Blockchain-based electronic voting systems have a bright future ahead of them, but overcoming current obstacles will require strategic planning and teamwork. Investigating hybrid models that incorporate elements of blockchain and traditional models is an important direction for future research. By implementing blockchain technology gradually into current electoral systems, this phased approach helps to allay concerns about its cost and complexity. To ensure a comprehensive understanding of the socio-political implications of blockchain adoption in the context of e-voting, interdisciplinary collaboration between computer scientists, political scientists, and policymakers is also imperative.

Furthermore, the establishment of interoperability protocols and standardisation are essential to building a global framework for blockchain-based e-voting[9]. Creating uniform guidelines can speed up international cooperation and the adoption process for countries looking to integrate blockchain technology into their voting processes. When it comes to organising activities and creating a cooperative atmosphere for the creation and application of safe and open electronic voting systems, international organisations can be extremely important.

In conclusion, the integration of blockchain technology into electronic voting systems represents a major advancement toward a more secure, efficient, and transparent democratic process. The potential advantages for election integrity and civic engagement are too great to be disregarded, but the difficulties listed above must be addressed with diligence and creativity. As we negotiate the difficulties of this emerging sector, a comprehensive understanding of how blockchain can impact electronic voting on a global scale in the future will be facilitated by the synthesis of research findings, case studies, and forward-looking recommendations.

2.1 Comparison of Literature Review

Name	Year	Author	Technique/Approach Used
E-voting System Based on Ethereum Blockchain Technology Using Ganache and Remix Environments	2023	Hassan et al. [14]	•Used Web3.js API, deployed using Ethereum smart contracts. •MetaMask was used as a wallet on a website, and Remix was used to deploy the smart contract on the main network. •Results show that the cost of each transaction is not stable, its increases with the increase in the network load, and the throughput ends up at 14 transactions per second.
Decentralised and Immutable E-Voting System using Blockchain	ICSCSS, 2023	Ashish Balti, Abhishek Prabhu, Sankar Shahi, Shrutika Dahifale, Dr. Vrajesh Maheta [15]	• Emphasis on Cryptography • Uses SHA 256 • Deterministic, Fast Computation • Identity is checked. • Provides an Admin Section to manage the system.

Secure and Decentralised Method of E-Voting using Blockchain and Smart Contracts	ICICCS, 2023	Kirit Enuga, Pranay Batthula, Sai Shashank Aleti, Nitish Kolluru, Sakthidharan G.R. [16]	Absolutely Same As Above
Blockchain based online E-voting System	2023, ICSCA	Youssef Abdelrahman Fekry Ali [17]	• Hybrid Approach. • Used ML along with Blockchain. • Uses Private Blockchain implementation. • Uses AI for Facial Recognition for verification. • HAAR Face Detector is used. • Uses SQL DB, for security.
Designing a Blockchain Enabled Methodology for Secure Online Voting System	2023, IDCIoT	Saurabh Singh, Alisha Singh, Shivam Verma, Rajendra Kumar Dwivedi [18]	• Layer by Layer model used. • Different nodes and contracts for different layers of voting.
Blockchain Based Secure Voting Mechanism underlying 5G Network: A Smart Contract Approach	2023, Research Article	Sachi Chaudhary, Shail Shah, Riya Kakkar [19]	• Uses Echidna Tool • Uses Public and Private Cryptography • Over Focus on 5G tech
ACB-Vote: Efficient, Flexible, and Privacy- Preserving Blockchain-Based Score Voting With Anonymously Convertible Ballots	2023, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 18	Wenyi Xue, Yang Yang et al. [20]	• Uses BBS+ signature, Homomorphic Encryptions • Anonymously Convertible Ballots • Uses CLS (Convertibly Linkable Signatures). • Uses a score(rating based) voting system
Blockchain based E-voting System	2019	Albin Benny, Aparna Ashok Kumar, Abdul Basit, Betina Cherian and Amol Kharat [21]	• Used Ethereum and Solidity. • Created a private net using Ganache. (Local Blockchain) • Used Ether.js API. • Hybrid system b/w Registration and Voting. • Used Biometric and Unique ID at Polling Booth to verify identity. • Created Hash based Login for One Time vote casting per login ID. Hash also provides Voter Privacy.

2.2 KEY GAPS IN THE LITERATURE

2.2 Key Gaps in literature Review

Author	Year	Purpose	Pros/ Cons
Kumar et al. [22]	2023	Blockchain and smart contract-based e- voting system	Transparent and authenticated. Ignored the scalability, latency and cost-related challenges
Zhu et al. [23]	2022	Multi-district elections based on blockchain-enabled e-voting	Authentication provided to all citizens for their votes, proper counting of votes through division in different layers.
Kaveri et al. [24]	2022	Reliable e-voting system with the use of blockchain	Easy to verify votes, open system. rational in decision making, smarter and reliable than traditional systems. Threat to system as update and changing of votes feature enabled, not secure
Lalitha et al. [25]	2022	Blockchain-based decentralized online voting mechanism	Voting from any place, authentication through Aadhar card, tamper-proof, provides election outcomes quickly, reduces manual cost and provides higher accuracy in counting. Some chances of vote tampering persist, not cost-effective
Naidu et al. [26]	2022	Blockchain and homomorphic encryption for protected voting	Secure against data manipulation and tampering . No discussion on response time and latency.

Farooq et al. [27]	2022	Transparent voting system using blockchain technology	Reduces injustice during voting, reliable, trans-parent, secure voting transactions Not perfect to be implemented on a large scale

The body of research on blockchain-based electronic voting systems points to a number of significant holes that need to be filled. First off, there aren't many thorough studies that address the scalability issues related to using blockchain in scenarios involving large-scale voting. Although blockchain provides security and transparency, it is yet unclear how well it will manage the large amount of transactions associated with national elections or referendums. Second, further study is required to fully understand how blockchain-based electronic voting systems work in terms of usability and user experience. To ensure widespread acceptance, it is imperative to comprehend how voters especially those who are not aware with blockchain technology interact with the system.

Furthermore, there is not enough research done in the literature on the legal and regulatory frameworks required to enable the use of blockchain technology in electronic voting, including concerns about identity verification, privacy, and the legitimacy of votes cast using blockchain technology. To advance the creation and adoption of reliable and inclusive blockchain-based electronic voting systems, these gaps must be filled.

Chapter 3: SYSTEM DEVELOPMENT

3.1 REQUIREMENTS AND ANALYSIS

This section introduces to the various requirements, backgrounds and preliminaries and also analyses them for a better user understanding. Requirements is a broad subject to discuss, and can be divided into basically two sub sections, Functional Requirements and Non-Functional Requirements.

3.1.1 FUNCTIONAL REQUIREMENTS

1. A Server to host ReactJS application.
2. An Account/Wallet to deploy Ethereum Smart Contracts with a reasonable Ether Balance.
3. Access to Firebase (Database and Authentication).

3.1.2 NON-FUNCTIONAL REQUIREMENTS

1. Performance: The program needs to operate smoothly and offer a good user experience.
2. Security: User data must be shielded from unauthorised access by the software.
3. Scalability: The program needs to be scalable in order to manage a big volume of data and users.
4. Reliability: The program needs to always be accessible and dependable.
5. Usability: The program needs to be simple to use and straightforward to navigate.

3.1.3 TECHNOLOGIES USED

1. Ethereum Blockchain
2. ReactJS
3. Firebase
4. Bootstrap
5. web3.js

3.1.4 BACKGROUND AND PRELIMINARIES

Other than the technical requirements, in order to understand the project we have to get a basic understanding of the background and preliminaries. These refer to the topics that are necessary to be understood in order to have an in depth understanding of the project.

1. **Secret Ballot:** A blockchain-based electronic voting system's secret ballot ensures voters' privacy and anonymity. Every voter in this virtual system is given a distinct cryptographic key, which enables them to securely cast a ballot without disclosing who they are. This key prevents vote manipulation and coercion by generating a secure digital signature. After that, the vote is encrypted and put to the blockchain, guaranteeing voter privacy and transparency. By enabling people to voice their opinions without worrying about the consequences, this method promotes voter confidence and upholds democracy's core values.

2. **Elliptic Curve Cryptography (ECC) Encryption:** In blockchain-based electronic voting systems, ECC encryption is crucial for enhancing the security and privacy of the election process. Because elliptic curve cryptography (ECC), a type of public-key cryptography, leverages the mathematical characteristics of elliptic curves to offer robust security with shorter key lengths, it is particularly well-suited for resource-efficient contexts like blockchain networks. It ensures the anonymity of the voting process and restricts access to the vote to just the corresponding private key by permitting the encryption of votes using the recipient's public key. Furthermore, ECC's capacity to generate digital signatures contributes to the integrity of the chain of transactions by guaranteeing the authenticity of votes.

3. **AES Encryption:** Strong and extensively used symmetric encryption, Advanced Encryption Standard (AES) 256 encryption is renowned for its extraordinary security and adaptability. AES 256, created by the National Institute of Standards and Technology (NIST), uses a 256-bit key size and works with 128-bit data blocks. Because of the enormous number of possible key combinations ensured by this level of encryption strength, it is extremely resistant to brute-force attacks. Multiple rounds of

substitution and permutation operations are used in the substitution-permutation network (SPN) structure that underpins AES 256 encryption. The algorithm's resilience to complex cryptographic assaults and its ability to operate well on a variety of computing platforms are its key strengths.

AES 256 is now widely accepted as the industry standard for encryption across a wide range of sectors. It is used to protect sensitive data in a variety of applications, including financial transactions, communication protocols, and government communications. Its incorporation into international encryption standards, which guarantee compatibility and interoperability across many systems and platforms, serves to further support its acceptance. AES 256 encryption is a mainstay of contemporary cryptography techniques and is essential for protecting data integrity and secrecy in an increasingly digital and networked environment.

4. **Cryptographic Hash Functions:** A key component in guaranteeing the security and integrity of digital data is the use of cryptographic hash functions. These mathematical formulas accept any size of input data and output a fixed-length character string known as a digest or hash value. Cryptographic hash functions are characterised by their one-way nature, which makes it computationally impossible to reverse the process and extract the original input from the hash.

   Hash functions in cryptography are traditionally a one way encryption process, where the data is divided into blocks of bits and encrypted for the use. The uniformity, fixed output sizes, collision resistance properties and pre-image resistance makes it a great protocol to ensure the validity and verifiability of the data without exposing the information contained in the data. Although there are now protocols to create "two way hash functions" or TWHF, our use case is best suited with the "collision resistant hash functions" or CRHF.

   By definition CRHF must have two inherent properties (second pre-image resistance and collision resistance) and one additional property (pre-image resistance) :

1. **Pre-image Resistance:** If we have an input x to a hash function $h(x)$, which produces an output y, it is computationally infeasible to compute x, which fulfils the criteria $y = h(x)$, given only y.

2. **Second Pre-image Resistance:** If we have been given an input x, it is computationally infeasible to compute an input supposedly x' that can fulfil the criteria that $h(x) = h(x'), x \neq x'$.

3. **Collision Resistance:** It is also computationally infeasible to produce two different inputs x and x', which can fulfil the criteria $h(x) = h(x')$.

5. **Blockchain:** Beyond just supporting digital currencies, blockchain is a ground-breaking decentralised technology with many uses. It is the technology behind cryptocurrencies like Bitcoin. Fundamentally, a blockchain comprises a distributed ledger that securely, transparently, and impenetrably documents transactions conducted among a group of computers. The chain forms a continuous and irreversible sequence because each block holds a cryptographic hash of the one before it. Because of this design, data immutability is guaranteed, which boosts security and trust across a range of businesses. Because blockchain technology is decentralised, it does not require middlemen, which lowers the possibility of fraud and increases efficiency. It is used in voting systems, supply chain management, and healthcare in addition to finance. Its usefulness is further increased by smart contracts, which are self-executing agreements with stipulations directly encoded into code. Blockchain is a game-changer, transforming conventional procedures and ushering in a new era of open and untrustworthy communication.

6. **Homomorphic Encryption:** A revolutionary method in cryptography, homomorphic encryption allows computations on encrypted data without requiring decryption. When using standard encryption, data must first be decrypted in order to be used, which puts it at risk for security issues. On the other hand, homomorphic encryption maintains the confidentiality of encrypted data by enabling calculations to be done directly on it.

This development has important ramifications for data processing and cloud computing privacy. Even when sensitive data is outsourced for processing, confidentiality can be maintained via homomorphic encryption. It reduces privacy issues by enabling users to assign calculations to outside servers without disclosing the underlying data.

There are two types of homomorphic encryption: fully homomorphic and partially homomorphic. Fully homomorphic encryption is a more flexible method for intricate computations than partially homomorphic encryption, which only permits addition or multiplication operations.

Although homomorphic encryption is a strong tool for computation that protects privacy, it has computational drawbacks and frequently needs a lot of processing resources. Its efficiency is being optimised for real-world applications through ongoing research and development, and it has the potential to completely transform secure data processing and outsourcing across a range of industries, including financial services and healthcare.

7. **Score Voting:** Voters use Score Voting, sometimes referred to as Range Voting, an easy-to-understand election system in which they rank several candidates according to their preferences. A numerical score is assigned to each candidate.

   Range voting uses a straightforward mathematical method to calculate the winner based on voter-assigned scores. Let's examine a situation in which there are N candidates and each voter can give each one a score from 0 and M.

   The total of each candidate's scores is the mathematical formula used to determine the winner of the Score Voting:
   The candidate with the greatest cumulative score is proclaimed the victor after the total scores for each candidate have been determined.

Total Score for Candidate $i = \sum_{j=1}^{V} Score_{ij}$

Because of its clarity and simplicity, Score Voting is transparent and simple to use. Voters can express their preferences by giving candidates scores, and the candidate with the greatest overall score—that is, the one with the broadest appeal—wins.

3.2 PROJECT DESIGN AND ARCHITECTURE

A blockchain-based voting system's architecture and design seek to address the shortcomings of conventional voting procedures by offering an unalterable, transparent, and safe platform for elections. The project makes use of blockchain technology to create a decentralised system that is resistant to fraud, improve transparency, and guarantee the integrity of votes.

In this section we try to explain the design and working of the framework using flowcharts and diagrams.

3.2.1 SECURITY MANAGEMENT OF THE FRAMEWORK

The security levels of our system can be divided on basis of the mentioned phases in IV.I. Hence, the management of security can be analysed on a few distinct levels, like Client-Side security, framework design security and data exchange security.

3.2.1.1 CLIENT SIDE SECURITY FOR FAIR ELECTIONS

At the client side part of things, we are able to observe that however efficient the system is there are things like the balancing of security, functionality and usability which acts as a trade-off contract amongst the features of the application. This can be explained by the Figure 4.5.

The Figure 4.5 shows a triangle whose vertices resemble Security, Functionality and Ease of Use of an application. The goal is to find a point ‘A’ which defines the features of the

application, and lies exactly at the circumcenter of the triangle. If not, the application would be facing trade-offs amongst the three.

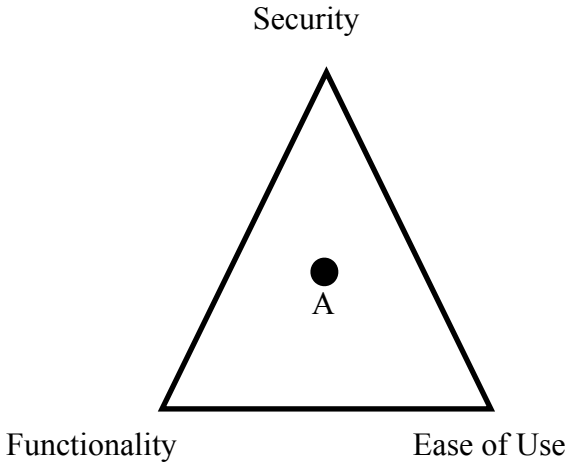

Figure 3.1 Security, Functionality and Usability Trade-off

The logical functioning is similar to that of a CIA triad. Protecting user data and reducing threats require a high level of security, but putting strong security measures in place can be more complicated and require more development work. To make sure that the program satisfies user needs while reducing vulnerabilities and guarding against cyber threats, it is imperative to strike a balance between security, functionality, and convenience of use. While having more features increases an application's usefulness and value, having too many features might complicate things and increase security risks. While prioritising simplicity of use increases user pleasure and engagement, doing so at the expense of security features or oversimplifying can put consumers at risk. Delivering an application that is secure, functional, and easy to use requires striking a balance between these factors.

Therefore, some features that our framework uses, at some cases, might not be the most functional option and other times it might not be the utmost level of secure. We have tried our best to analyse most of the relevant literature and previous implementations to pick up the most optimum features for our framework. For context, all the security features are analysed independently in the subsequent sections.

This trade-off triangle is also the reason for many counter questions that may be asked. For example, “Why did we use UID for user identification if we already have the hashed voter ID available to us?”. It is solely because of finding the previously mention point ‘A’, where it functionally and security-wise better to use both in their different contexts.

Another problem that can only be addressed using this analogy, is “Why to ask the user to vote physically when blocked (due to unauthorised or suspicious activity) when our framework is a software?”. Again, the answer is in finding the point ‘A’ of balanced tradeoff. Since, the requirement is of high integrity in such a framework that can potentially decide the future of a large amount of population. Therefore, It is essential to restrict the users that may have an intention of performing unauthorised activities and on the other hand also not debar them from utilising their voting rights just on the basis of a software’s suspicion.

3.2.1.2 SECURITY FOR DATA EXCHANGE AND CHOOSING THE OPTIMAL HASHING ALGORITHM

For the security that has been implemented for the data exchange part of the framework, we have used hash functions to transmit the data that does not need to be decoded like the data which comprises mostly of the user data, especially the voter identity number, is needed only for verification and validity purposes. Voter identity number is necessary to be kept track of because it helps maintain the integrity of the votes that have been casted and consequently the integrity of the election conducted. Another property of a valid Voter-ID number is that it is unique to every individual and is issued by the authorities that are conducting the elections, therefore it can be verified by cross referencing the data.

A drawback that could be faced while using Voter-ID is the need to expose it because the vote casted will be deployed on the blockchain. This practice of exposing the identity number of the voter to anyone can create heavy setbacks regarding to the integrity of the elections. Since, the Voter-ID is of such importance, but still cannot be exposed publicly,

hashing can ensure that Voter ID number is verified without the need to expose the identity of the voter itself.

To find the most optimal hash function for our use case we compared the three most secured and relevant hashing protocols of the date, namely Two Way Hash Function (TWHF), One Way Hash Function (OWHF) and Collision Resistant Hash Function (CRHF). Out of these, TWHF [18], is eliminated at first because of the need of a one way encryption, hence complicating the process and using extra computational resources by using this process is not necessary.

OWHF and CRHF [20] can be compared with relevance to this study. As mentioned in Menezes' Handbook of Applied Cryptography [20], OWHF is a weak hash function when compared to CRHF. The three properties [19] used to compare the hash functions as already discussed in section III.III of this study, are pre-image resistance, second pre-image resistance and collision resistance. Out of these three, a classic OWHF only fulfils pre-image resistance [20]. On the other hand, CRHF was designed to fulfil collision resistance feature. It is evident that collision resistance implies second pre-image resistance of hash function. Additionally, the property of pre-image resistance can be added to a collision resistant hash function, which can then fulfil all the three properties. Hence, with the properties of pre-image and collision resistance in CRHF, it is considered to be the most optimal protocol for our use case.

The SHA-256 could be an optimal way to go for, as it provides both pre-image and collision resistance. On comparing it to SHA-512, it has been found that it gives a 50% performance increase over an identical application of SHA-256 [26]. SHA-256 can even be truncated with SHA-512 for even better performance and storage advantages [26]. Therefore this hash function is the most optimum to use for our framework.

3.2.1.3 CHOOSING THE OPTIMAL ENCRYPTION ALGORITHM

There are multiple encryption algorithms available today to use, and many of these are very advanced in true nature, like the DHIES implementation based on the Diffie-Hellman problem [24]. Our use case, though in need of such secured encryption schemes, will most likely be suited with an asymmetric encryption algorithm. It is consequent to the fact that the use case is in data exchange between two different parts of the system.

Therefore, the most basic explanation of our case is to use a function on the front-end framework, $Enc(x, pubkey)$ where, x is the data provided by the user which is basically the votes casted, and $pubkey$ refers to a public encryption key. $Enc(x, pubkey)$ encrypts the data x using the public key and the algorithm predefined.

During the verification and validity of votes, the function $dec(y, privkey)$ is used, where $dec()$ is a function over the already encrypted data y, which decrypts the data using the private key $privkey$ and the predefined decryption algorithm.

The presented encryption framework is unique even after using the same asymmetric algorithms. It is because most of the cases where such encryption is used, a new public private key pair is generated for each user, whereas in this case we used a singular pair, making the public key completely public which is possible due to multiple levels of authentication done previously to this phase. The private key is held only by the election organiser and should be used only when the votes are to be read and counted.

It is a quite a well known fact that ECC and RSA are two of the most secured encryption algorithms as of date. When it comes to efficiency, ECC is almost on par with RSA. Smaller key sizes like the ones used in ECC lead to quicker cryptographic operations and lower storage needs, which improves the efficiency of ECC in resource-constrained situations such as IoT (Internet of Things) devices and mobile devices. On the other hand constant and very low encryption times on RSA make it a tough competitor to ECC in terms of efficiency. In the same context, other algorithms lay far behind in the comparative analysis like, in order to achieve comparable degrees of security, methods such as AES and

others usually require substantially bigger key sizes, which results in additional computational overhead and storage requirements.

3.2.1.4 Elliptical Curve Cryptography VS RSA

With today's technological advancements, it is a heavy task to decide the best suited algorithm for the use case. The decision between the Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC) encryption algorithms has been closely examined in the fields of classical and quantum computing. While both ECC and RSA are essential components of contemporary cryptographic systems, there are notable differences between their methods and effectiveness, especially when comparing how well they function in situations with classical and quantum computing.

3.2.1.4.1 COMPARISON IN CLASSICAL COMPUTING

When it comes to computational efficiency and key size, ECC outperforms RSA in classical computing [30]. ECC achieves security levels that are comparable to those of RSA by using reduced key sizes. For example, even if a 256-bit ECC key offers the same level of security as a 3072-bit RSA key, ECC offers faster cryptographic operations and lower computing overhead due to its shorter key lengths.

In classical computing contexts, ECC has been shown to offer many advantages over RSA in numerous research investigations. Notably, a thorough investigation carried out by Johnson et al. [29] showed how effective ECC is on devices with limited resources and how it can reduce computational overhead in large-scale cryptography operations.

It is also noted in research that RSA takes a constant time to encrypt data, which is close to 0 seconds with today's technology, and ECC shows an increase in encryption time with more security bits [30], but the total time taken for encryption and decryption was lower with ECC. In the same study, in terms of operational effectiveness and security with fewer parameters, it was concluded that ECC performs better than RSA and that devices having resource constraints are better suited with ECC [30].

3.2.1.4.2 COMPARISON IN QUANTUM COMPUTING

Quantum computers have the ability to attack both RSA and ECC. Shor's algorithm, which effectively factorises big integers and cracks RSA encryption, can break RSA encryption. Similarly, Grover's technique can effectively half the effective key size of ECC, making it vulnerable to quantum attacks.

However, in reality, ECC is somewhat more immune to quantum attacks because it requires far smaller key sizes to attain the same level of security as RSA. Even though this is the case, it would not be a reasonable decision to replace RSA with ECC just because of this factor. ECC in a post-quantum scenario would not hold much respect for securing data because it will still be relatively easy to crack.

Therefore, it does not make much difference in the conclusion of the decision. It is only a matter of time that both of these and most of the other security algorithms(at present time) are relevant because of the fast progress in the quantum computing world. Therefore, a need for new security algorithms which are resistant to quantum attacks is present.

All things noted, it is slightly better to use ECC for our use than a higher key sized RSA. Even though not much better, but a slightly more efficient way to be used for our framework would be ECC encryption.

3.2.2 USER EXPERIENCE WORKFLOW

Figure 3.2 demonstrates how the user i.e. voter, will go about the voting system. Important stages like registration of the voter's identity, approval to vote for the given identity and checking whether the voter has already submitted the vote or if the voter is banned from participating in the election, are shown in this flowchart from the voter's perspective.

The user will login using the Gmail account. A Gmail account is necessary because of the additional security that we are getting from it. The user email is hidden because Gmail provides an authentication token which contains a hash value for each unique user, by the key name of "UID".

The application then checks if the user has been registered by the Governing Authority, if No, The user gets redirected to a registration portal where the user has to input the Voter ID details, which are first stored in a temporary database, which is verified by the governing authority. When the VoterID is verified by the governing authority, we know for sure that User is a valid user and is eligible to vote. In case the VoterID details are found invalid or the authority finds that the voter is ineligible to vote, the Voter's Gmail Account is deregistered from our server and is blocked. The voter is then logged out, and cannot access our portal again.

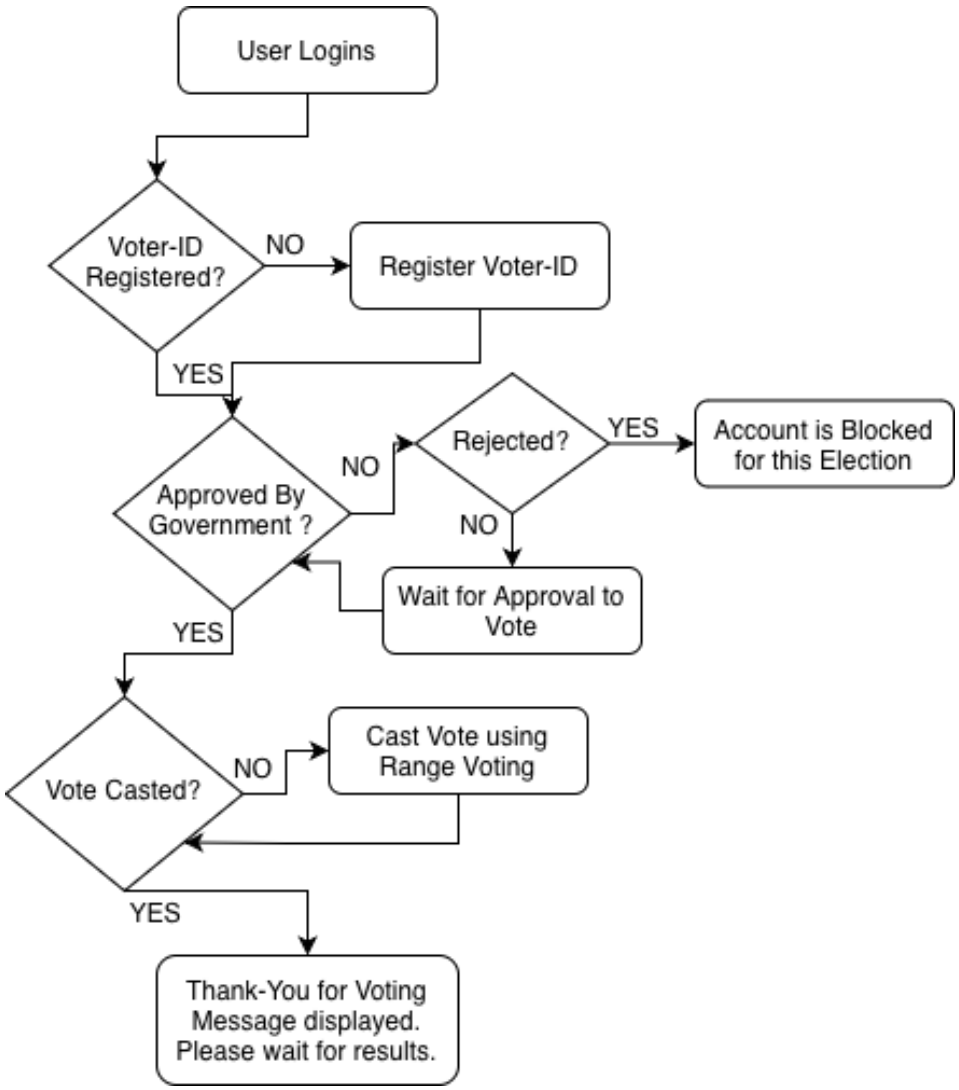

Figure 3.2 User Experience Workflow

When the Voter is finally registered, we now allow the voter to vote in a score voting manner. The scores are given to the candidates and as soon as the voter submits the score. The data envelope is encrypted and sent to the blockchain to make it immutable and in a way, public information.

3.2.3 ENCRYPTION FRAMEWORK FLOW CHART

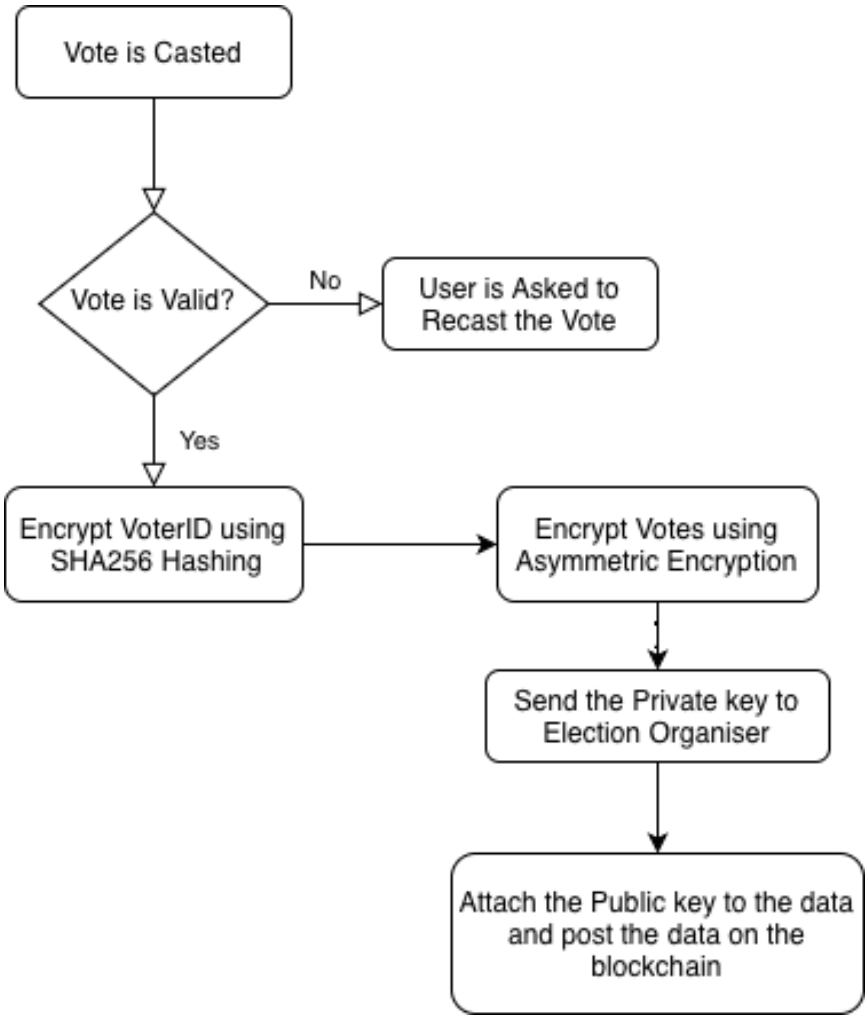

Figure 3.3 Encryption Framework

Figure 3.3 shows how the data envelope is created and encrypted in its entirety.

The process starts when the vote is casted. If the vote is not a valid vote, the user is simply redirected and is asked to vote again. In case it is a valid vote, the VoterID is encrypted

using SHA256 Hashing Algorithm. Along side the VoterID, the votes are encrypted too. The Votes are encrypted using Asymmetric Encryption techniques. The Private Key is sent directly to the election organiser, and the public key is attached to the data envelope.

The Data Envelope is sent to the smart contract running the blockchain related code, and is subsequently posted to the ledger. This makes the data immutable and also hidden due to the encryption techniques used and absence of the private key. The election organiser can make the private key publicly available in case anyone wants to check the integrity of the election results.

3.2.4 VOTE VERIFICATION ALGORITHM FLOW CHART

Figure 3.4 explains the working of the vote verification algorithm and how this blockchain-based voting system's vote verification algorithm guarantees the fairness and openness of the election process.

The verification function is called after the election has ended. The Election organiser will be given the right to do so. This function can also be a very handy tool when a third party requests to verify the Voting System.

The function first checks the duplicate votes, if one VoterID hash has been linked to multiple votes, the algorithm has been programmed to only keep track of the latest vote casted and discard all the previous votes. The votes are then averaged out and stored as per the score voting rules.

This vote verification algorithm offers a clear and safe procedure that allows voters to confirm the authenticity of their votes while protecting the privacy of individual selections by combining public-key infrastructure, cryptography, and smart contracts. The system is further strengthened by ongoing audits and adherence to best practices in blockchain security and cryptography.

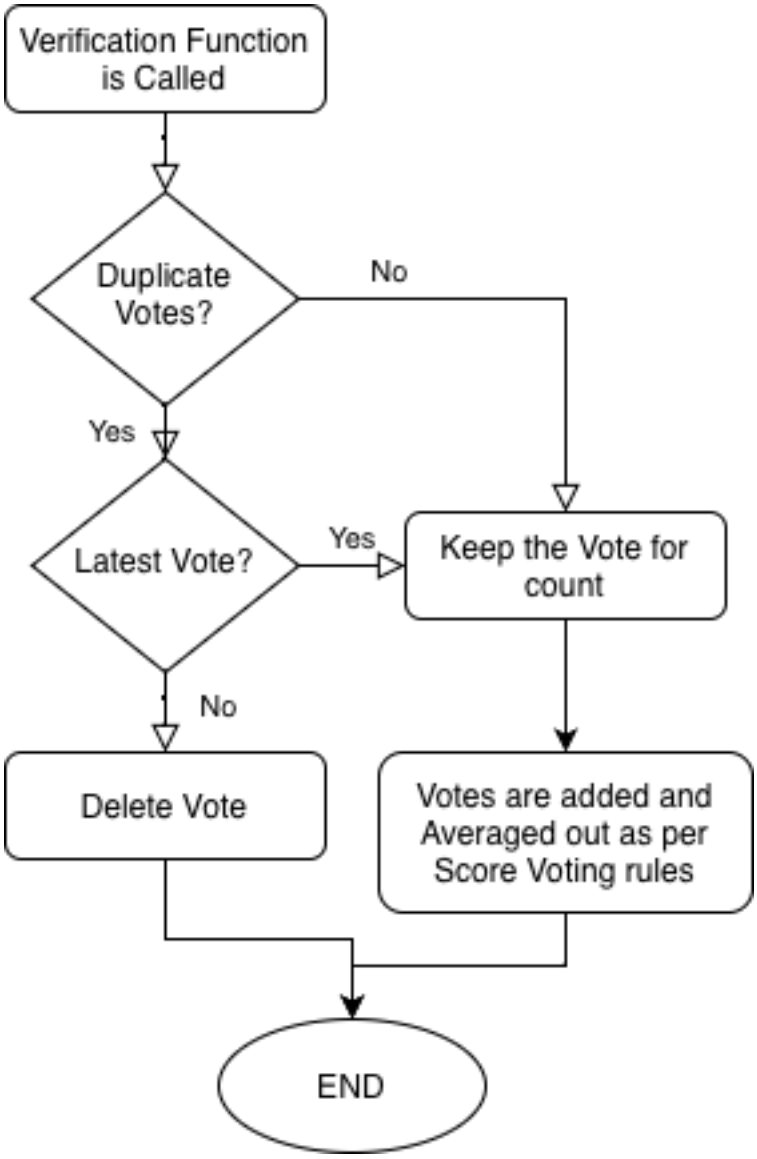

Figure 3.4 Vote Verification Algorithm

The architecture and design of this project lay the groundwork for a safe, open, and effective blockchain-based voting system. The solution guarantees voting process integrity and offers an easy-to-use interface for both election managers and voters by utilising the potential of blockchain technology and smart contracts. The system's dependability is enhanced by scalability controls and routine security assessments, which make it a strong option for updating election procedures.

3.3 DIVISION OF PHASES

For a streamlined and manageable workflow from our(developer's) perspective, we have divided the whole project into multiple phases. This division made us easier to tackle niche problems and develop the algorithm in a better and more efficient way.

3.3.1 PHASE I

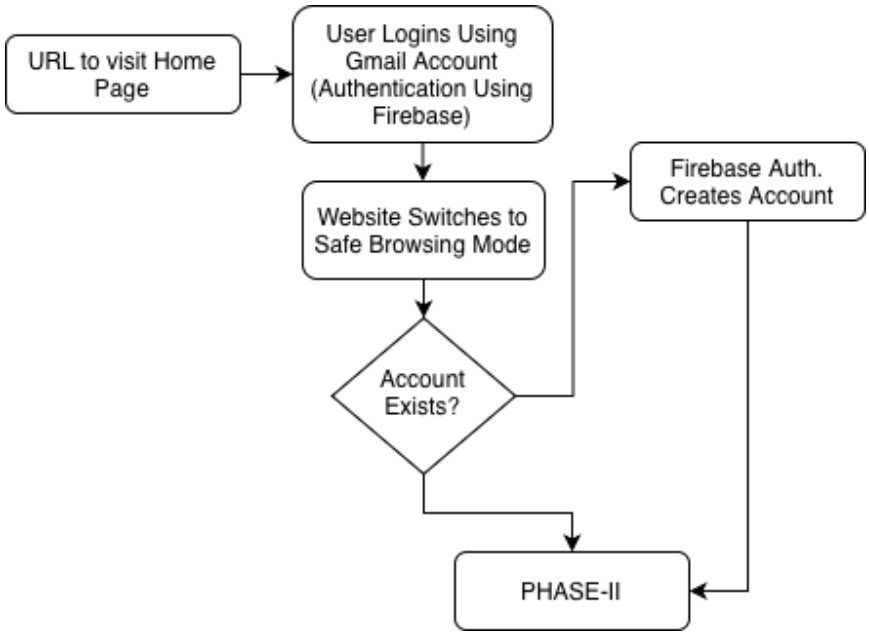

Figure 3.5 PHASE I

Figure 3.5 demonstrates the insights of the initial stage of the framework. The voter visits the home page of the website from where the first stage of authentication starts, using the Google Auth provided by Firebase. The website after logging the user in, switches to the safe browsing mode, and then carries on the other authentication tasks and then proceeds to Phase II. Phase I is supposed to be an entry phase for the framework.

3.3.2 PHASE II

Figure 3.6 showcases the flow of the website after the user is initially authenticated by Google Auth and the application has gone into safe browsing mode. This safe browsing can be achieved through a variety of approaches like, adding browser extensions which prevent

user from accessing certain features or even implementing something like a special browser which prevents actions like screenshots, tab switching, shortcut key combinations etc. and only works in full screen mode.

The Phase II involves authenticating the voter by cross checking it's details with the governing authority's data. Voter's identity is checked by verification of a voter identity number. During the whole time that the voter spends on the web application, and even after the voter logs out, the voter is identified by this voter identity number. To make sure that this number does not get compromised, it is client-side encrypted using state of the art hashing algorithms, in our case, SHA-256. Once, the voter's identity is approved and verified, the voter can move on to Phase III for final voting.

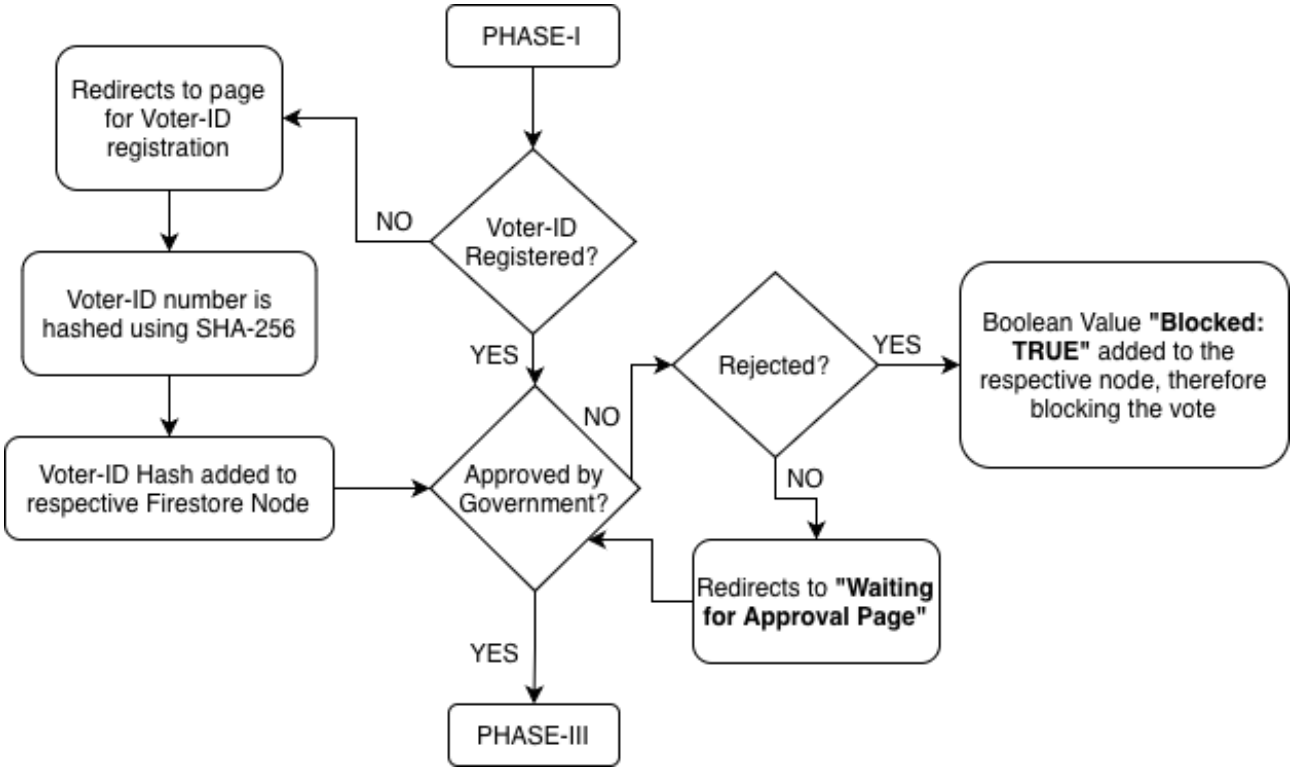

Figure 3.6 PHASE II

3.3.3 PHASE III

Phase III as shown in Figure 3.7, showcases the framework on how the user will be directed once it is registered. The first checking point being the question, "Has the voter

already casted the vote?". If yes, the voter is redirected to eventually log out from the application and is prevented from casting another vote. If not, the voter is redirected to a vote casting page, where a range voting system has been set up to allow the voters to flexibly cast their votes according to their wants.

The encryption scheme used to encrypt the votes is RSA-4096. A detailed explanation and comparison of this scheme with other alternatives especially ECC is discussed in the latter part of this paper. The votes after encryption are finally deployed on the blockchain which will make them immutable and final.

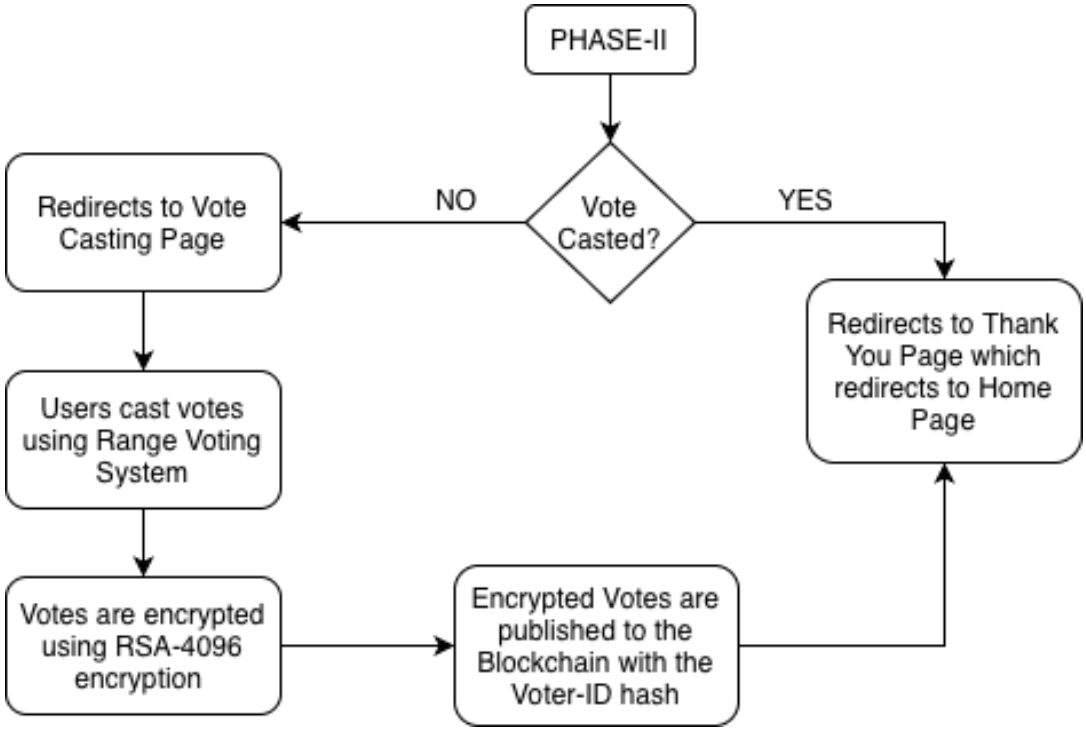

Figure 3.7 PHASE III

3.4 PSEUDO CODE

Here we present a complete, end-to-end pseudocode for the framework algorithm that is presented in the previously mentioned phases.

Begin

Phase 1: Let *U* be the user.

Step 1: Initial authentication.

```
SignUp(U){

// Uses Firebase internal functions to authenticate
and signup/login the user

return uid;

}

Uid = SignUp(U)
```

Phase 2: Further authentication for maximum security.

Step 1: Check for authority verified valid identity proof.

```
if(is_Identified_On_Database(U) == false){

id = Input("Enter Valid ID Proof and necessary
details: ");

}

else { // proceed to next step}
```

Step 2: Securing the identity proof.

```
hashed_id = hash(id)

post_to_db(hashed_id);

// hash() function will convert the identity number
to a secured standard hash which will then be
posted to the database.
```

Step 3: Validating the identity proof by the competent authority.

```
if(validate(hashed_id)==false){block_user(uid);}

// if user cannot be validated, that user ID will
be blocked for impersonation and will be asked to
vote physically.

else {// proceed to next step}
```

Phase 3: Voting.

Step 1: Check if the user has already casted the vote.

```
if(vote_casted(hashed_id))

{ // skip next step }

else{ // proceed to next step}
```

Step 2: Ask user to cast the vote.

```
// redirect user to the vote casting page

redirect(uid, "/voteCastingPage");

// Ask the user to cast votes and store them

votes = cast_vote_in_range_voting(uid);
```

Step 3: Encrypt the votes.

```
// Use of RSA-4096 to encrypt the voting data

// Transform the data(votes) in plaintext into an
integer V such that 0 <= V < n.

// Where, n = p * q, where p and q are distinct
prime numbers, each approximately 2048 bits in
length.

// Make use of the encryption function to calculate
the encrypted_data E: E ≡ m^e mod n

encrypted_data = encrypt(votes, public_key);
```

Step 4: Publish the votes

```
// Publish the votes on the blockchain alongside the
hashed voter identity.

publish_to_Blockchain(hashed_id, encrypted_data);
```

End

3.5 IMPLEMENTATION

This section describes the concrete actions needed to implement the proposed blockchain-based electronic voting system. The method of implementation takes a comprehensive approach, encompassing the creation of smart contracts, user interfaces, blockchain network connectivity, and the integration of strict security measures.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract EVotingSystem {
    address public admin;  // Address of the contract administrator
    address[] public candidates;  // Array to store candidate addresses
    mapping(address => uint256) public votes;  // Mapping to store votes for each candidate
    bool public votingOpen;  // Flag to indicate whether voting is open

    constructor() {     665984 gas 616800 gas
        admin = msg.sender;  // Set the contract creator as the administrator
        votingOpen = true;  // Voting is initially open
    }

    modifier onlyAdmin() {
        require(msg.sender == admin, "Only the administrator can perform this action.");
        _;
    }

    modifier onlyDuringVoting() {
        require(votingOpen, "Voting is closed.");
        _;
    }

    modifier onlyValidCandidate(address candidate) {
        require(isValidCandidate(candidate), "Invalid candidate address.");
        _;
    }

    function addCandidate(address candidate) public onlyAdmin {     51246 gas
        candidates.push(candidate);
    }

    function closeVoting() public onlyAdmin {     26600 gas
        votingOpen = false;
    }

    function isValidCandidate(address candidate) public view returns (bool) {     infinite gas
        for (uint i = 0; i < candidates.length; i++) {
            if (candidates[i] == candidate) {
                return true;
            }
        }
        return false;
    }

    function vote(address candidate) public onlyDuringVoting onlyValidCandidate(candidate) {     infinite gas
        require(votes[msg.sender] == 0, "You can only vote once.");
        votes[msg.sender] = 1;
        votes[candidate] += 1;
    }

    function getVoteCount(address candidate) public view onlyValidCandidate(candidate) returns (uint256) {     infinite gas
        return votes[candidate];
    }

    function getCandidateCount() public view returns (uint256) {     2446 gas
        return candidates.length;
    }

    function isVotingOpen() public view returns (bool) {     2589 gas
        return votingOpen;
    }
```

Figure 3.8 Smart Contract Code

Figure 3.8 is the actual code for our implementation of this project.

The code shows the algorithm for the smart contract. This contract is designed to be deployed on the Ethereum Blockchain, using an actual account. With respect to our project we found ways to use dummy Blockchains and Dummy Accounts, to test our code.

```
function getMessageEncoding(rating) {
  const enc = new TextEncoder();
  return enc.encode(rating);
}

function encryptMessage(rating) {
  const encoded = getMessageEncoding(rating);
  // iv will be needed for decryption
  const iv = window.crypto.getRandomValues(new Uint8Array(16));
  console.log("iv => " + iv);
  //   const iv = new Uint8Array(123456123456);
  window.crypto.subtle
    .generateKey(
      {
        name: "AES-GCM",
        length: 256,
      },
      true,
      ["encrypt", "decrypt"]
    )
    .then((key) => {
      console.log("key => " + JSON.stringify(key));
      return window.crypto.subtle.encrypt(
        { name: "AES-CBC", iv: iv },
        key,
        encoded
      );
    });
}
```

Figure 3.9 Code to Encrypt to AES

In order to encrypt data using AES-256 in JavaScript, one must usually use a cryptography library, such as CryptoJS, the Web Crypto API, or both.

Figure 3.9 shows our implementation of the encryption algorithm.

We used the WebCrypto API which is a standard API supported in modern browsers for cryptography.

Explanation:

1. An AES-256 random key that can be used for encryption and decryption is produced using the **generateKey** function.

2. The generated key and the data to be encrypted are passed to the encryptData function. It makes use of the Initialisation Vector (IV) for additional security in the AES-GCM method.

3. The sample usage shows how to encrypt a piece of data and produce a key. A binary array containing both the encrypted data and the IV is the end product.

Because of the asynchronous nature of the Web Crypto API, promises must be handled carefully. Therefore, it was very crucial for us to work and test our code in various environments and against various different test cases.

```
// Catch Rating value
const handleRating = (rate: number) => {
  setRating(rate);
  console.log(rate);
};

const handleSubmit = async (e) => {
  e.preventDefault();
  console.log("Submitted Rating: " + rating);

  const encryptedRating = encryptMessage(
    //   new Uint8Array(654321654321),
    rating
  );
  console.log(encryptedRating);

  await updateDoc(doc(db, "Voters", localStorage.getItem("uid")), {
    VoteForManmohanSingh: rating,
  });
};
```

Figure 3.10 Code to Catch Rating

Figure 3.10 shows the code to catch the rating value from the user.

One of the commented lines shows the use of “Uint8Array()”. It has been commented out for testing purposes.

An array of 8-bit unsigned integers is represented by the Uint8Array object. When working with binary data—that is, reading and modifying files, interacting with network protocols, and managing raw data buffers—it is especially helpful.

When working with binary data, this kind of array comes in very useful because it gives you more exact control over how to manipulate and interpret byte values.

There is also a type of this called “Uint16Array”. The Uint16Array is a typed array in JavaScript that symbolises an array of 16-bit unsigned integers.

```
// Hashing the Voter ID
const hashValue = (val) =>
  crypto.subtle
    .digest("SHA-256", new TextEncoder("utf-8").encode(val))
    .then((h) => {
      let hexes = [],
        view = new DataView(h);
      for (let i = 0; i < view.byteLength; i += 4)
        hexes.push((VoterID + view.getUint32(i).toString(16)).slice(-8));
      return hexes.join("");
    });

const VoterIDHash = await hashValue(
  JSON.stringify({ a: "a", b: [1, 2, 3, 4], foo: { c: "bar" } })
);

console.log("VoterID Hash =>  " + VoterIDHash);
```

Figure 3.11 Code to Hash VoterID

Creating a fixed-size hash value (256 bits, or 64 hexadecimal characters) from the input data (voter ID in this case) is the process of hashing a voter ID using the SHA-256 algorithm. This procedure guarantees that the hashed output is resistant to collision attacks, irreversible, and unique to the particular input.

It's crucial to remember that hashing voter identification numbers or any other sensitive data is standard procedure for system security and privacy, particularly in situations like computerised voting systems. The system can hash the voter ID that is provided and compare it with the stored hash for authentication during verification without disclosing the original voter ID. The hashed values can be safely saved.

One of the most important steps toward upgrading and changing electoral processes is the deployment of the blockchain-based electronic voting system. The system is positioned as a strong and cutting-edge solution due to the careful attention to security, transparency, and

user experience. As the landscape changes quickly, iterative development, testing, and continuous cooperation are still crucial to guaranteeing the efficacy and adaptability of the system.

3.6 INTEGRATION OF RSA-4096

As we read more literature and gained more experience with this project, we decided to switch from AES to RSA-4096 for encryption. There are a number of strong arguments for switching from the previous AES implementation in this system to RSA 4096 encryption, the main ones being security, compatibility, and key management.

Above all, because RSA 4096 is more resistant to cryptographic assaults than AES, it provides a better degree of security. The 4096-bit key length offers a substantially larger key space with RSA's asymmetric encryption scheme, which still uses a pair of keys (public and private) for encryption and decryption. This makes it computationally impossible for adversaries to decrypt encrypted data using brute force or other methods. This improved security is especially important for applications such as electronic voting systems, where it is critical to preserve the integrity and confidentiality of sensitive voter data in order to retain public confidence in the electoral process.

The inclusion of RSA 4096 might address compatibility problems with the earlier AES implementation. Although AES is widely used and efficient for symmetric encryption in many scenarios, it can occasionally be challenging to integrate it with certain systems or platforms, especially when those systems or platforms need to be compatible with pre-existing infrastructure. On the other hand, RSA encryption is a fundamental cryptographic technique that is extensively supported by a variety of cryptographic libraries and systems, ensuring seamless integration and interoperability within your e-voting ecosystem.

Additionally, secure key management processes are made simpler to execute by the asymmetric nature of RSA, particularly in scenarios where safe key exchange or distribution is essential. When employing RSA encryption, the private keys required for decryption are kept private while the public keys required for encryption are freely shared.

This asymmetric key management technique reduces the likelihood of unauthorised access to sensitive data and streamlines the logistics of key distribution, enhancing the overall security posture of your electronic voting system.

```
const RSAencryption = (rating) => {
  var publicKey = forge.pki.publicKeyFromPem(
    `-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAkQrSaiJyy0MsyZlTyWjQ
gbxvdGoCUlWjWJph/B6f91h0q9/CEzSP9c8lTUs5B5inGAGy/7U5N+eRgKXYYi81
UUoQEyzujPwHkltaj4O3vZrx2+H1L8wuHRhlP+kMVUkfOovT/N2ECvp59E44+8u5
/orbJNQUJP9HOHxb4Dl0VeJVSmMUt+I8KvAhJRb7xZuNnZrTOLrw4p75NuERDo+t
QlykPtx0dfBIKTsnNwykd22Q0XnpgfiEVmyC021sCOsIGZlkbFase/PlTHJucVzp
Wy9Fby2WhF2Prtfz4Ybnf7StyJKYdQQokKRUEJcEGj6zB82MNGcgqy4VUgoKGORc
Oen8k1LCQHiZiWmnXOOxn4AUGTbm3ZAPy6GIGaQFOYo58sY0YIsZcSOGTRWIrr9V
/gDABnkzZnzJ/bymlx565QOqXFvYJP773E4aP0GAFk9lNKBzVsIEn4VYxKGxUnQV
8UocChuRRGflBgPkW8CtbMyIwls9A4NlZSquxB+v1r3PCcG+gt/BSi9+vvwFkJRM
84exq/wrqLxR0qLorkVDmBXeVRRAmrEUI+mC+bLJGuiKfXn5/V2tPGuwd7QrBfMr
7bqStCzTAyJRcZxtk4QtEFSLwG0CxDOM3+6Ic0AnZo8zRmN3h8mX3Nd3ymw58Jxp
NraFT9a0hnhwVm9TcviGVXUCAwEAAQ==
-----END PUBLIC KEY-----`
  );

  const encryptedRating = forge.util.encode64(
    publicKey.encrypt(forge.util.encodeUtf8(rating), "RSA-OAEP", {
      md: forge.md.sha256.create(),
    })
  );
  console.log("enc: " + encryptedRating);
```

Figure 3.12 RSA-4096 Implementation Code

In this code it is easily observable that we have used a common public-private key pair for all the data flowing through that route. This was a strategic move to increase efficiency, by not regenerating the public private key pairs every time for every vote for every user. For an electronic voting system, it could make sense to strategically decide to maintain the RSA public key available and accessible to all users in order to optimise system efficiency, user happiness, and transparency.

One way to save computational cost and simplify key management is to utilize a single RSA public key for all users. If each user had their own set of RSA public keys, key distribution and generation would need additional resources and would be more difficult to manage. By employing a shared public key, you may encrypt data uniformly for each user by simplifying the encryption process and doing away with the requirement for distinct keys.

Maintaining consistency in the RSA public key makes it easier for users to obtain and use. Users no longer have to worry about obtaining and maintaining their own public keys, which speeds up the onboarding process and reduces the chance of user error. This method enhances user experience and encourages wider adoption of the electronic voting system by removing unnecessary obstacles from the voting process for users.

The RSA public key is kept shared and available to all users in order to promote the concept of transparency in electronic voting systems. Election accountability and transparency are promoted by ensuring that all encrypted data is accessible and verifiable by everyone using a shared public key. This openness, which increases the overall integrity of the electronic voting system and gives insight into the encryption process, promotes confidence among stakeholders, including voters, election organisers, and regulatory agencies.

3.7 KEY CHALLENGES

A blockchain-based electronic voting system presents a number of issues that must be carefully considered in order to guarantee the integrity, security, and broad acceptance of the system. Identification and remediation of a variety of difficulties, from security flaws to worries about regulatory compliance, have defined the implementation process. The development of a robust and flexible electronic voting system has been greatly aided by the recognition and resolution of these issues.

Among the principal difficulties we faced are:

1. **Vulnerabilities**: Resolving any openings for malevolent actors to exploit in the blockchain network and smart contracts.

2. **Duplicate Voting**: Using strong identification verification procedures to stop cases of identity fraud or duplicate voting.

3. **Finding the Correct Balance Between Transparency and Privacy**: Ensuring voter anonymity while allowing for result verification while striking the proper balance between transparency in the voting process and protecting voter privacy.

4. **Transaction Volume**: Increasing the blockchain network's capacity to manage a lot of transactions at peak voting periods without sacrificing security or speed.

5. **Data storage**: Keeping track of the blockchain's growing size as more votes and users sign up over time.

6. **User Experience**: Making sure voters, particularly those who are not familiar with blockchain technology, have a smooth and easy-to-use experience.

7. **Digital literacy and accessibility,** concerns for a broad spectrum of voters, including individuals with impairments, are addressed.

8. **Legal standards**: Modifying the electronic voting system to conform to current election laws and standards.

9. **Standardisation**: Handling the lack of uniform laws governing blockchain-based electronic voting, which may differ in different states.

10. **Public Perception**: Reducing mistrust and increasing public confidence in the new technology, especially in areas where there is opposition or a lack of knowledge about blockchain.

11. **Code flaws**: To find and fix any flaws that can jeopardise the voting process's integrity, smart contracts should undergo extensive audits on a regular basis.

12. **Ensuring trustworthy and safe procedures** for voter identity verification while maintaining anonymity is known as "secure authentication."

13. **Protecting Against Sybil Attacks**: Defending against Sybil attacks, in which a malicious party fabricates several false identities in order to tamper with the election results.

14. **Resolving issues** with the blockchain-based electronic voting system's integration with current electoral infrastructure and government databases.

15. **Cross-Platform Compatibility**: Making sure that various systems work together to promote wider adoption.

16. **Public Understanding:** To boost acceptability and participation, educating the public about the advantages, features, and security protocols of the blockchain-based electronic voting system is important.

A multidisciplinary strategy incorporating technology, regulatory frameworks, and public participation is needed to overcome these obstacles. It takes teamwork, flexibility, and ongoing research to successfully deploy blockchain-based electronic voting systems.

Chapter 4: TESTING

In order to guarantee the dependability, security, and compliance with legal requirements of a blockchain-based electronic voting system, testing is an essential stage of development. The testing methods, important test cases, and testing phase results are described in this chapter.

4.1 TESTING STRATEGY

1. **Functional Testing:** User Registration: Check that all aspects of voter registration, such as identity verification and key issuing, are carried out correctly.

2. **Voting Procedure:** Verify that the votes are cast and recorded accurately and in accordance with the blockchain protocol. Verify the operation of the smart contracts that control voter registration, ballot production, and vote tallying.

3. **Identity Verification in Security Testing:** Examine how reliable identity verification systems are in order to stop unwanted access.

4. **Vote encryption and decryption:** Assess the robustness of the cryptographic algorithms utilised.

5. **Consensus Algorithm**: Evaluate the consensus algorithm's resistance to possible intrusions.

6. **Usability Testing:** User Interfaces: Evaluate how easy it is for voters, election officials, and administrators to use the interfaces.

7. **Accessibility:** Make sure the system can handle users with different degrees of digital knowledge and needs.

8. **Cross-Platform Compatibility:** Examine how well the system works on other platforms.

4.2 TEST CASES AND OUTCOMES

The testing phase verifies the performance, security, usability, and regulatory compliance of the blockchain-based electronic voting system. For the same reason, we applied the following test cases, and received the mentioned outcomes.

4.2.1 TEST CASES

1. **User Registration:** Check that only qualified voters are able to successfully register as users. Verify that a distinct cryptographic key is provided to every voter who has registered.

2. **Voting Process:** Examine the blockchain's vote recording accuracy. Make sure that just the specified voting period is used by voters to cast ballots.

3. **Smart Contracts:** Verify that voting rules are appropriately executed using smart contracts. Verify that votes are counted using smart contracts automatically and accurately.

4. **Identity Verification:** Evaluate how well identity verification systems work against false identities. Make that the voting system is only accessible to voters who have been verified.

5. **Encryption and Decryption:** Examine how reliable the cryptographic algorithms that are used to encode and decrypt votes are. Make certain that votes are kept private at all times.

6. **Regulatory Compliance:** Verify that the electronic voting system complies with all applicable laws and regulations. Make sure that requirements for privacy are upheld without sacrificing adherence to regulations.

4.2.2 OUTCOMES

1. Successful voter registration, key issuance, and identity validation are the functional testing outcomes.
2. Smart contracts provide accurate voting tallying and recording.
3. Results of security testing: Strong identity verification systems that block unwanted access.
4. Robust encryption and decryption techniques protect votes from manipulation.
5. Results of usability testing: Interfaces that are easy to use for all user types.
6. Features for accessibility meeting a range of user requirements.
7. Results of performance testing: Scalability of the system at different transaction loads.
8. Votes cast during the voting period are recorded in real time.
9. Results of integration testing: Easy integration with current databases and systems. Interoperability across multiple platforms and devices.

The favourable results point to a stable and dependable technology that is prepared for use in actual election procedures. It is advised to conduct frequent audits and continuous testing to handle changing security risks and preserve the integrity of the system over time.

Chapter 5: RESULTS AND EVALUATION

Although the field of study on blockchain-based electronic voting systems is still in its infancy, first findings point to both encouraging developments and persistent difficulties. Positively, it has been shown that blockchain technology can improve the security and openness of electronic voting procedures. The blockchain's decentralised structure and immutability can greatly lower the dangers of fraud and manipulation. But there are still issues, mainly with usability, scalability, and connection with current legal systems. When blockchain is used for large-scale elections, scalability problems occur, which raises questions about how effectively transactions are processed.

5.1 RESULTS

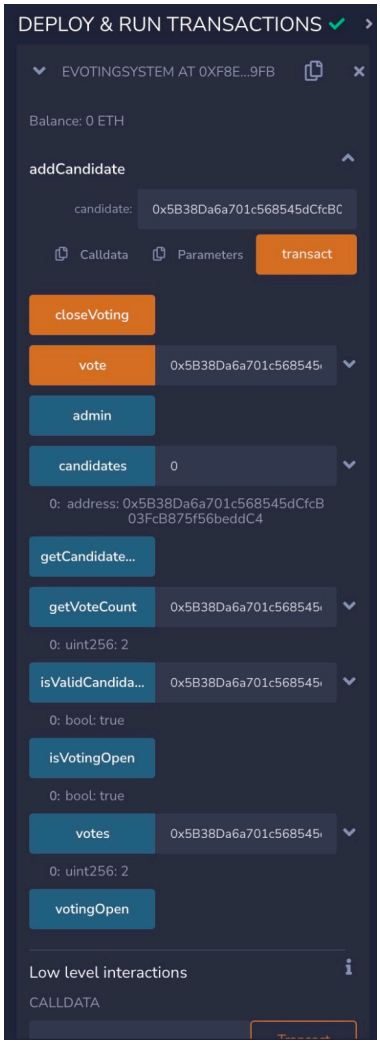

Figure 4.1 Smart Contract Results

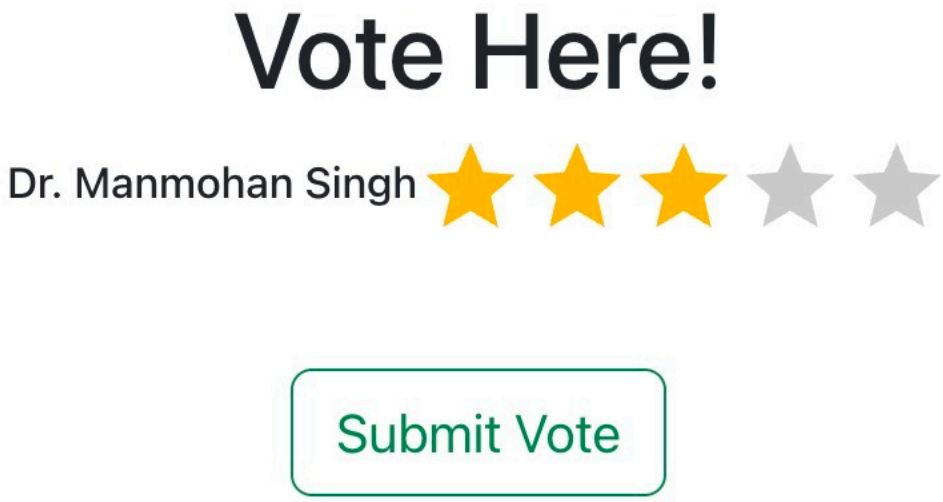

Figure 4.2 Score Voting Interface

A voting site that uses range voting, or score voting, is an example of a contemporary, inclusive democratic procedure. With this approach, voters are not restricted to selecting just one candidate; instead, they give each contender a numerical score that represents their preferences. Voters can indicate how strongly they support or oppose each candidate by assigning a score within a predetermined range, usually ranging from 0 to a maximum value, on the portal, which normally displays a list of candidates.

```
VoterID Hash =>                                                    Register.jsx:55
44aa106279f5977f59f9067fa9712afc4aedc6f5862a8defc34552d8c7206393
```

Figure 4.3 Generated VoterID hash

Voters can express their preferences more accurately and capture the nuances of their views on various candidates with this sophisticated approach. The candidate who receives the highest total score after all votes are cast is declared the winner. By encouraging openness

and voter expression, score voting creates a system that more accurately represents the range of viewpoints held by voters. This voting technique could discourage strategic voting and encourage people to cast their ballots honestly without worrying that their selections won't be used. With its user-friendly interfaces and clear instructions, the voting site acts as a platform that makes it easier for voters to engage in a more complex and expressive political process.

Data security and integrity are vitally dependent on the commonly used cryptographic hash function known as SHA-256 (Secure Hash Algorithm 256-bit). The method of creating a SHA-256 hash starts with choosing the message or data input to hash. This input can be any length, ranging from a straightforward text passage to a more intricate collection of data. Subsequently, the algorithm divides the input into fixed-size blocks, each of which is handled in turn. SHA-256 performs several rounds of bitwise operations, mathematical transformations, and logical operations on each block.

The algorithm generates a 256-bit hash value iteratively by combining the input data with a series of constant values and initial hash values, or "IVs," One of SHA-256's most important characteristics is its resilience to collision attacks, which means that it is computationally impossible for two distinct inputs to result in the same hash output. The avalanche effect of the hash function is exacerbated by the fact that even a slight alteration to the input data produces a significantly different hash. Because of the deterministic nature of the SHA-256 algorithm, the same input will always result in the same hash output. In a variety of applications, including cryptographic methods, the generated hash acts as a distinct digital fingerprint for the original data, offering a safe and effective way to confirm data integrity and validity.

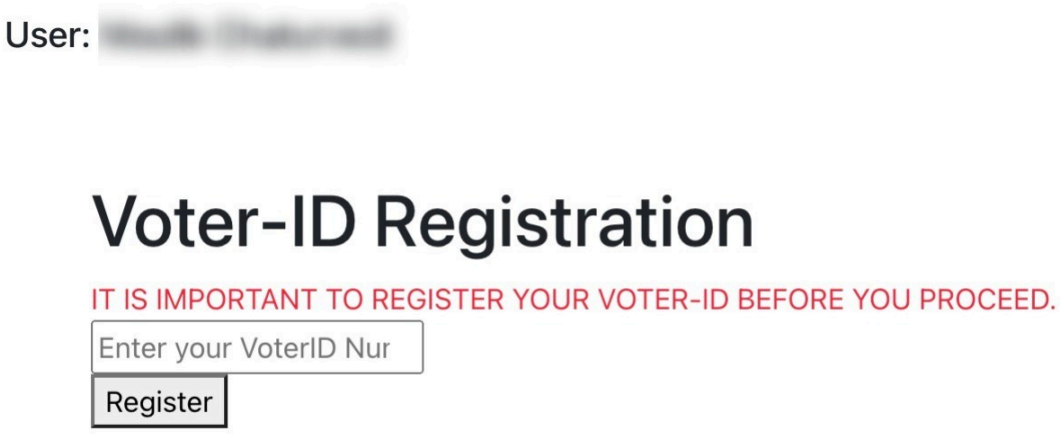

Figure 4.4 VoterID Registration Portal

Voter ID registration portals are online platforms created to make voter registration easier by giving qualified individuals an easy way to apply for voter identity cards. Election authorities or government entities in charge of monitoring the electoral process usually administer this web platform.

The gateway starts by confirming a person's eligibility to register to vote. To find out if they match the requirements for voter registration, users must submit personal information such as name, date of birth, residence, and citizenship status. Frequently, applications ask applicants to upload supporting documentation attesting to their citizenship, residency, and identity. A legitimate government-issued ID, evidence of residency (utility bills, rental agreements), and any other pertinent documents designated by election officials may be included in this set of documents.

Securing the personal data that is provided during the registration process requires strong security protocols. This covers safe data storage, encryption techniques, and defence against unwanted access.

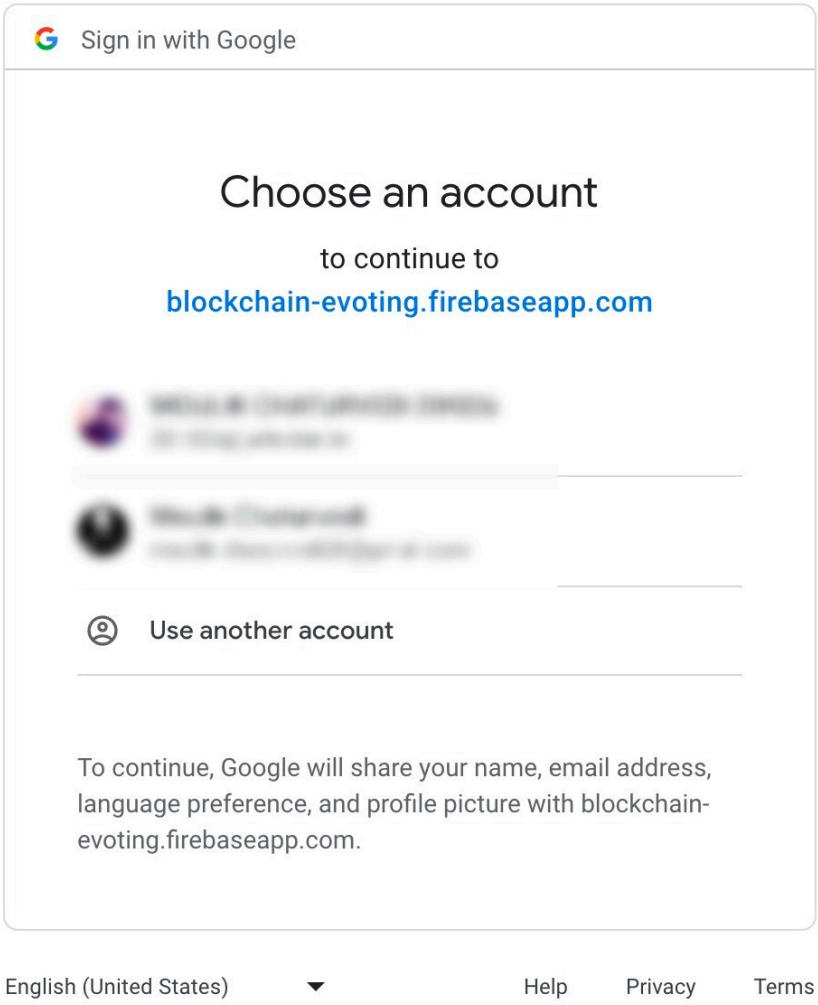

Figure 4.5 Account Login

Developers can quickly integrate user authentication into their online and mobile applications with Firebase Authentication, a tool offered by Google's Firebase platform. Numerous authentication techniques are supported, including phone number, email and password, and social identity providers like Google, Facebook, Twitter, and others.

5.2 COMPARISON WITH EXISTING SOLUTIONS

This section assesses the suggested blockchain-based electronic voting system against current options, taking into account each one's advantages, disadvantages, and general efficacy in resolving the issues with conventional voting procedures. The goal of the comparative analysis is to draw attention to the unique qualities and benefits that the blockchain-based electronic voting system offers to the election process.

When compared to current solutions—traditional voting systems, electronic voting without blockchain, and distant online voting—the suggested blockchain-based e-voting system is positioned as a revolutionary development. Though they are well-known, traditional voting systems frequently have fraud vulnerabilities and inefficient manual procedures. On the other hand, while blockchain-free electronic voting methods are faster, they still present issues with transparency and centralised databases. While remote online voting increases accessibility, it also poses security problems and makes voter anonymity difficult to maintain.

The comparison of SHA-256 with Google Authenticator and RSA (Rivest-Shamir-Adleman) in the framework of current blockchain-based electronic voting systems raises several important points. The popular cryptographic algorithm RSA provides a tried-and-true option with robust security characteristics and broad interoperability across cryptographic libraries and systems. Stakeholders are reassured about the integrity of the electronic voting process by its dependability and simplicity of integration into blockchain networks. But the greater key lengths of RSA might result in processing overhead, which could affect how quickly transactions happen on the blockchain.

The e-voting system acquires an extra layer of security when Google Authenticator is used in conjunction with SHA-256 encryption for two-factor authentication. SHA-256, which is renowned for its resilience and ability to withstand collisions, guarantees the privacy and accuracy of information kept on the blockchain. By adding a second authentication layer, the integration of Google Authenticator improves user verification and reduces the possibility of unwanted access to the electronic voting platform. In order to increase

confidence and transparency in the election process, SHA-256 and Google Authenticator work together to strengthen the security foundation of blockchain-based electronic voting systems.

The strong security features of the blockchain-based electronic voting system set it apart. It creates a transparent and tamper-resistant ledger by utilising decentralised blockchain technology and cryptographic procedures, which reduces the vulnerability to fraud found in traditional and some electronic systems. Because the system produces an unchangeable record, it encourages accountability and gives voters the ability to independently confirm the veracity of the results. This is in sharp contrast to traditional systems, where there is frequently a compromise in transparency, and electronic systems that do not utilise blockchain technology would not be able to offer the same degree of verifiability.

A key component of the blockchain-based electronic voting system is privacy concerns. Through the use of cryptographic techniques, the system balances secrecy and openness while preserving voter anonymity. On the other hand, during the counting and verification procedures, voter privacy may unintentionally be jeopardised by conventional and some computerised systems. This issue is well addressed by the suggested approach, which also makes sure that the voting procedure is kept private and secure.

Another area in which the blockchain-based electronic voting system excels is accessibility. It allows for remote voting and adds accessibility features without sacrificing security, providing a solution for people with mobility issues. By contrast, conventional methods could be inaccessible to people with disabilities, and while distant online voting is possible, security precautions must be carefully considered.

The blockchain-powered electronic voting system sticks out as a complete remedy for the drawbacks of the current voting techniques. It offers a strong substitute for conventional and electronic voting methods due to its decentralised, transparent, and secure characteristics, which improve the electoral process's integrity. The advantages of enhanced

security, transparency, and accessibility make the blockchain-based electronic voting system a revolutionary invention that has the power to influence democratic practices in the future, despite certain obstacles. To properly develop and use this technology, cooperation, testing, and ongoing research will be essential.

Chapter 6: CONCLUSION

A ground-breaking and inventive solution to the problem of updating electoral processes is the blockchain-based electronic voting system. This in-depth paper has walked readers through the ideation, creation, testing, and deployment of a system that is ready to take on the problems that come with using conventional voting techniques. Through the use of strong smart contracts, decentralised blockchain technology, and cutting-edge cryptographic techniques, the suggested solution promotes accessibility, security, and openness in the electoral sphere.

The comparison with current methods highlights the unique benefits of the blockchain-based electronic voting system. It is notable for its ability to withstand fraud, voting process openness, voter privacy protection, and improved accessibility. This approach not only resolves long-standing problems but also lays the groundwork for a more secure and inclusive electoral future as it leads the way in a paradigm shift in democratic practices.

There are constant hurdles in the complex field of cybersecurity, and the blockchain-based electronic voting system is not exempt. Thorough testing, frequent security audits, and working with specialists to find and fix vulnerabilities were given top priority during the implementation phase. One of the mainstays of the system's resistance to possible attacks is the ongoing development of security measures.

The practical procedures used to bring the conceptual idea to life are detailed in the implementation chapter, which also covers user interfaces, security protocols, and blockchain network architecture. The system has undergone a rigorous process of testing, iteration, and adaption in order to meet the demands of deployment in the real world. The focus on user education and training guarantees administrators, election officials, and voters a seamless transition.

During the implementation phase, a complex web of cryptographic protocols, consensus processes, and user-friendly interfaces has been woven together to produce a reliable

system that resolves issues with traditional voting systems that have existed for a long time. The focus on security measures, such as frequent security audits and sophisticated cryptographic algorithms, guarantees the voting process's integrity while reducing potential risks and reassuring stakeholders of the system's dependability.

A fundamental component of the blockchain philosophy, transparency is ingrained in the entire system. An auditable record of each transaction is made possible by the immutable ledger, giving the voting process a level of transparency never before seen. Election results published on the blockchain give voters the option to independently confirm the results' correctness while also improving accountability.

Proactive mitigation solutions have been implemented to address challenges such as regulatory compliance, user acceptance obstacles, and security risks. The system's robustness is continuously improved through the use of an iterative development approach, regular security assessments, and cooperation with legal experts.

It is critical to recognise that the blockchain-based electronic voting system is a dynamic solution that will need to change in response to societal demands, security considerations, and technological improvements. It is advised to conduct ongoing research, testing, and iterative development to improve the flexibility and resilience of the system.

6.1 FUTURE WORKS AND POSSIBILITIES

Due to the limited scope of our project, in terms of research area, time, and our limited expertise and experience, there is only so much that we know and can execute. We have tried our best to make this project up to date, equipped with the latest and most efficient technologies of our time. Still possibilities remain for a wide variety of upgrades that can be made in the future for this project.

For instance, due to our limited technical capabilities, we were unable to implement a full fledged end to end ECC encryption. It is noted and mentioned in this project report multiple times on how ECC is better than RSA at many instances.

Another important thing that might be taken into consideration in future is the advancement of quantum computing and its ability to break today's widely used security algorithms with ease. When the matter of discussion is of national importance, we cannot overlook this possibility. Therefore a need of integrating quantum-proof security algorithms is definitely of increasing demand.

As we approach to the end of our report, it is clear that the blockchain-based electronic voting system is a catalyst for a fundamental reinvention of democratic processes rather than just a technological innovation. Its incorporation of accessibility, security, and openness meets the changing demands of a digital society. For the system to stay at the forefront of democratic and technical growth, more cooperation, study, and adaptation are required. An important turning point in the evolution of democracy has been reached with the implementation of the blockchain-based electronic voting system, which paves the way for a day when inclusiveness and trust will be key components of election procedures.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 15th, May 2024

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report ✓ | Paper

Name: Moulik Chaturvedi & Garvita Sharma **Department:** Computer Science **Enrolment No** 201326 & 201122

Contact No. 8005060991 **E-mail.** moulik.chaturvedi26@gmail.com, 201122@juitsolan.in

Name of the Supervisor: Dr. Aman Sharma, Assistant Professor (SG), Computer Science & IT

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): BLOCKCHAIN BASED ELECTRONIC VOTING SYSTEM

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 72
- Total No. of Preliminary pages = 8
- Total No. of pages accommodate bibliography/references = 6

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 14 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
15th May, 2024	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String	14%	Word Counts	14421
Report Generated on			Character Counts	80641
15th May, 2024		**Submission ID**	Total Pages Scanned	72
		2379799901	File Size	6.9 MB

Checked by
Name & Signature **Librarian**

.....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Decentralized Bus. Rev., pp. 21260, Oct. 2008.

[2] Z. Zheng et al., "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in Proc. BigData Congress, 2017.

[3] P. Ehin et al., "Internet voting in Estonia 2005–2019: Evidence from eleven elections," Government Information Quarterly, vol. 39, no. 4, pp. 101718, 2022.

[4] S. Semenzin et al., "Blockchain-based application at a governmental level: disruption or illusion? The case of Estonia," Policy and Society, vol. 41, 2022.

[5] "Election-Hacking Lessons From the 2018 Def Con Hackers Conference," [Online].

[6] W. D. Smith, "Range Voting," [Online].

[7] S. Venugopalan et al., "BBB-Voting: 1-out-of-k Blockchain-Based Boardroom Voting," [Online].

[8] W. Xue et al., "ACB-Vote: Efficient, Flexible, and Privacy-Preserving Blockchain-Based Score Voting With Anonymously Convertible Ballots," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 3720-3734, 2023.

[9] Y. Yang et al., "PriScore: Blockchain-Based Self-Tallying Election System Supporting Score Voting," Trans. Info. For. Sec., vol. 16, pp. 4705–4720, 2021.

[10] P. McCorry et al., "A smart contract for boardroom voting with maximum voter privacy," in Proc. Int. Conf. Financial Cryptography Data Secur., 2017, pp. 357–375.

[11] X. Yang et al., "Blockchain voting: Publicly verifiable online voting protocol without trusted tallying authorities," Future Generation Computer Systems, vol. 112, 2020.

[12] "Ballot Stuffing" [Online].

[13] M. Balinski and R. Laraki, "A theory of measuring, electing, and ranking," Proc. Nat. Acad. Sci. USA, vol. 104, no. 21, pp. 8720–8725, May 2007.

[14] "Balinski Laraki's Orsay Range Voting Experiment,"

[15] "Independent Party of Oregon Makes History with New Voting Method," [Online].

[16] H. Devillez et al., "Can we cast a ballot as intended and be receipt free?," in IEEE Symposium on Security and Privacy 2024, May 2024.

[17] D. J. Moskowitz and J. C. Rogowski, "Ballot Reform, the Personal Vote, and Political Representation in the United States," British Journal of Political Science, vol. 54, no. 1, pp. 22-39, 2024.

[18] T. Koroglu and R. Samet, "Can There Be a Two Way Hash Function?," in IEEE Access, vol. 12, pp. 18358-18386, 2024.

[19] P. Rogaway and T. Shrimpton, "Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance," in Fast Software Encryption, Springer, Berlin, Heidelberg, 2004.

[20] A. Menezes et al., Handbook of Applied Cryptography. CRC Press, 1996.

[21] S. Wang et al., "An Overview of Smart Contract: Architecture, Applications, and Future Trends," in 2018 IEEE Intelligent Vehicles Symposium (IV), 2018, pp. 108-113.

[22] C. Nguyen et al., "Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities," IEEE Access, pp. 1-1, 2019.

[23] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," 2012.

[24] M. Abdalla et al., "DHIES: An encryption scheme based on the Diffie-Hellman Problem," 2001.

[25] J. Bevers, "The Study of Symmetric and Asymmetric Key Encryptions," 2021.

[26] S. Gueron et al., "SHA-512/256," 2010.

[27] A. Attaallah and R. Khan, "Estimating Usable-Security Through Hesitant Fuzzy Linguistic Term Sets Based Technique," Computers, Materials and Continua, vol. 70, pp. 5683-5705, 2022.

[28] D. Popescul, "The Confidentiality – Integrity – Accessibility Triad into the Knowledge Security. A Reassessment from the Point of View of the Knowledge Contribution to Innovation," 2011.

[29] S. Arumugam et al., "Analysis and Implementation of ECC Algorithm in Lightweight Device," in Proc. ICCSP, 2019.

[30] D. Mahto and D. Yadav, "RSA and ECC: A comparative analysis," International Journal of Applied Engineering Research, vol. 12, pp. 9053-9061, 2017.

[31] Different types of blockchain technologies. https://thebossmagazine.com/ different-types-of-blockchain-technologies/.

[32] What is Double Spending. https://corporatefinanceinstitute.com/resources/knowledge/other/double-spending/.

[33] What is Smart Contract. https://developers.rsk.co/guides/full-stack-dapp-on-rsk/part1-overview/.

[34] Introduction to Solidity. https://www.geeksforgeeks.org/introductionto-solidity/.

[35] NodeJS installation v14.17.5. https://nodejs.org/ko/blog/release/v14. 17.5/.

[36] Features Of Truffle Ethereum. https://www.edureka.co/blog/developingethereum-dapps-with-truffle/.

[37] What is Truffle Suite. https://www.upgrad.com/blog/what-is-trufflesuite/.

[38] Install Ganache. https://www.trufflesuite.com/ganache/.

[39] Web2 vs Web3. https://ethereum.org/en/developers/docs/web2- vsweb3/.

[40] MetaMask Chrome. https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn?hl=en/.

APPENDIX

GLOSSARY OF TERMS

Aadhaar verification: It is a system used in India for verifying someone's identity through authentication.

Accessibility: Refers to the capability of a system to be easily used by individuals with disabilities.

Blockchain: A technology that enables recording transactions on a computer network through distributed ledger.

Bottom up: Describes a system that is organised starting from the levels rather than being dictated from the top down.

Cryptographic techniques: Methods used for encrypting or decrypting information securely.

Decentralised voting system: A voting system that doesn't rely on an authority to oversee and manage the voting process.

Distributed ledger technology (DLT): A technology enabling tamper proof record keeping across multiple nodes in a distributed network.

Data integrity: Refers to ensuring the accuracy and completeness of data.

Electoral fraud: The act of tampering with the voting process in order to influence the outcome of an election

Error handling: The process of addressing and resolving errors that may occur during the voting process.

Hack attacks: Attempts made to gain access to computer systems or networks illegally.

Immutability: The property where something cannot be changed or altered once it's established or recorded.

Manipulation: The act of controlling or unfairly influencing something or someone.

Mobile app: A software app meant for use on a device.

Multi factor authentication; A security measure that asks for forms of identification to

gain access to a system.

Non profit organisations: Organisations that aren't run with profit as the goal but rather with other purposes, in mind.

Network simulation: A computer program that mimics the way a network behaves.

Online voting: Voting that takes place using the internet.

Regulatory Compliance: refers to the process of adhering to laws and rules.

Scalability: pertains to a systems capability to manage a growing volume of information or users.

Smart contracts: are contracts that execute automatically and are stored on a blockchain.

Tampering: denotes the alteration of data.

Transparency: refers to the quality of being visible or comprehensible, to others.

Verifiability: The ability to be checked or confirmed

WCAG 2.0: The Web Content Accessibility Guidelines (WCAG) 2.0 are a set of standards for making web content more accessible to people with disabilities.