A Blockchain Technology Based Voting System

Apurv Anand[1], Kamlesh Lakhwani[2], Shruti Mathur[3]

[1]M. Tech Student, Dept. of CSE, JECRC University, Jaipur, Rajasthan
[2]Professor, Dept. of CSE, JECRC University, Jaipur, Rajasthan
[3]Assistant Professor, Dept. of CSE, JECRC University, Jaipur, Rajasthan

Abstract

Enhancing electronic voting systems to ensure election security and integrity while maintaining cost-efficiency has been a focus of research for decades. Consequently, numerous technological advancements have been integrated into modern voting systems. However, the adoption of technology has introduced new challenges. These include the risk of inaccurate election results due to technical malfunctions, vulnerabilities to election security breaches, and the potential for tampering with results. Additionally, concerns persist about the ability to perform reliable post-election audits to verify and instill confidence in election outcomes.

This research explores the application of blockchain technology and zero-knowledge protocols to enhance the security, integrity, and transparency of electronic voting systems. A novel voting algorithm is proposed, designed to complement existing systems by providing verifiable evidence of election accuracy. A prototype implementation is developed, demonstrating the system's security and audit capabilities. The Rhode Island voting system serves as a case study for this investigation. The proposed algorithm is compatible with current election technologies and addresses key issues associated with current voting systems.

Keywords: Blockchain, Security, Decentralized, Digitalizing.

CHAPTER 1

Introduction

Electronic voting systems have been extensively studied for decades, with the primary aim of reducing election costs while ensuring security, integrity, and voter privacy. These efforts have led to significant advancements in voting technology, including optical ballot scanners, paperless systems, encrypted voting platforms, and internet-based voting systems.

Key concerns include the risk of inaccurate election outcomes due to technical failures, security vulnerabilities that could allow unauthorized alterations, and a lack of transparency in the electoral process. The inability to perform post-election audits also undermines public trust in election results. These challenges highlight the need for continued innovation and improvement in voting technologies.

Blockchain technology, introduced in 2008 by Satoshi Nakamoto, is best known for its role in supporting cryptocurrencies through public transaction ledgers. It functions as an open, decentralized ledger capable of recording transactions between parties in an efficient, verifiable, and permanent manner.

A blockchain's structure is designed to resist data manipulation. Once information is stored in a block, altering it retroactively requires modifying all subsequent blocks and obtaining consensus from the majority of the network. This design ensures the integrity of the stored data.

The key features of blockchain technology include:

- Decentralized storage: The ledger is replicated across multiple locations, making it resistant to tampering by altering data in a single location.
- Distributed control: The ability to add new transactions is shared among participants rather than controlled by a central authority.
- Immutable structure: Each new block references the previous state of the ledger, creating a secure and unchangeable chain.
- Consensus-based updates: Before any new block is permanently added, a majority of network participants must agree on its validity.

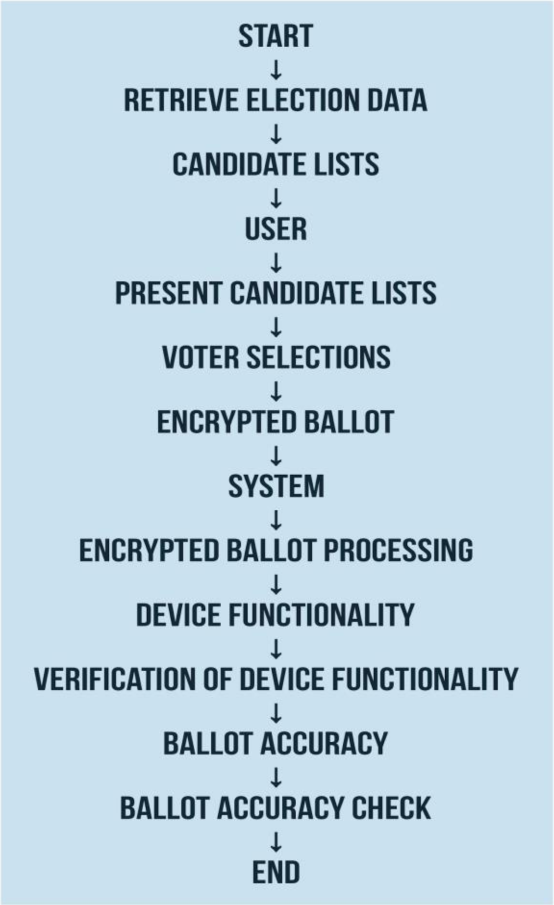


Figure 1. Electronic voting systems

CHAPTER 2

This chapter reviews key voting systems that have influenced the current research. It provides an overview of the ballot structures, voting processes, and auditing features of each system.

2.1. Voting Systems

1. Aperio:Aperio is a paper-based voting system that enables the creation of verifiable audit trails without relying on cryptographic methods. It is an end-to-end verification mechanism suitable for environments using secret paper ballots, without requiring advanced election machinery.
2. Scantegrity II:Scantegrity II is an end-to-end cryptographic voting system designed for optical scan technology. It uses invisible ink to print unique confirmation codes on the ballot, allowing voters to verify their choices securely.

3. Helios:Helios is an open-source system tailored for online elections. Built on public-key cryptography, it includes an auditing capability to ensure election integrity. Developed in 2008 by Ben Adida at MIT, Helios enables users to organize and conduct fully online elections.
4. WAVERI:WAVERI, short for "Watch, Audit, Verify Elections for Rhode Island," is an election algorithm grounded in set theory. It offers a solution for generating verifiable audit trails without the complexity of cryptographic techniques.

2.2. Election Auditing Methods & functions

1. Ballot Audits:A ballot audit ensures that ballots are printed accurately. On election day, voters or auditors can request a blank ballot from a poll worker for verification.
2. Receipt Audits:Receipt audits allow voters and monitoring groups to confirm that all ballot receipts are included in the final vote count. After the election, an auditing team collects receipts from voters and matches them with corresponding ballots, helping identify any missing ballots during the election process.
3. Tally Audits:Tally audits provide a way to verify the accuracy of the vote count. Different methods can be used depending on the voting system, but the primary objective is to confirm the correctness of the final tally.

2.3. Risk-Limiting Audits (RLAs)

Under this law, election officials must perform an RLA by examining a random sample of ballots selected through statistical modeling, rather than auditing a fixed number of ballots. This method enhances confidence in election results by using statistically sound techniques.

2.4. Blockchain Technology

The foundation of blockchain-like systems was first introduced in 1982 by cryptographer David Chaum in his dissertation titled “Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups.” This idea was expanded in 1991 by Stuart Haber and W. Scott Stornetta, who developed a cryptographically secure chain of blocks aimed at creating tamper-proof document timestamps.

In 2008, the modern concept of blockchain was introduced by an individual or group known as Satoshi Nakamoto. This design employed a Hashcash-like system to timestamp blocks without relying on a trusted intermediary. A year later, Nakamoto launched Bitcoin, a cryptocurrency that utilizes blockchain as a decentralized public ledger to record transactions.

2.5. Proof of Work

Proof-of-Work (PoW) is a consensus mechanism designed to prevent network abuse, such as denial-of-service attacks or spam. It requires service requesters to perform computational tasks, typically consuming processing power, to access a service.

2.6. Cryptographic Hash Functions:

A cryptographic hash function, or hashing algorithm, processes a block of data and transforms it into a fixed-size string known as a hash value. This transformation is deterministic and designed to scramble the input data securely. An essential feature of a reliable hash function is that it should be computationally infeasible for two different inputs to produce the same hash value.

CHAPTER 3

Litrature review

A Blockchain-Based Voting

This chapter introduces a new voting algorithm that leverages blockchain technology alongside cryptographic methods such as the El-Gamal encryption system and zero-knowledge protocols. This algorithm enhances security and supports comprehensive auditing, offering improvements over traditional voting systems.

3.1. The Voting Process:On election day, each voter is given an unmarked paper ballot, free of identifying information,This ensures compatibility with the state's current optical scanning devices, avoiding the need for reconfiguration.Voters mark their selections on the ballot as per usual. Once completed, the ballot is inserted into a scanner, which validates it for errors such as overvotes or incorrect markings.

3.2. Validator Authentication Process: Before a vote is recorded in the blockchain, a validator generates a block hash through a process known as "proof of work." Each hash must meet specific formatting criteria, achieved by using a random value called a nonce. The nonce is generated individually for each ballot by a secure central workstation, referred to as the "central server." Validators do not store nonces locally, requesting them from the central server for each ballot to maintain security.

To ensure secure communication, validators must authenticate themselves to the central server using the Fiat-Shamir zero-knowledge protocol. This method enables validators to prove their identity without transmitting sensitive data, thereby preventing interception and unauthorized access during the authentication process.

3.3. Proof of Work:Proof of Work (PoW) is a key mechanism used to secure the integrity of data stored in the local blockchain. It prevents unauthorized alterations by embedding the hash value of the chain's previous state into each new block before it becomes part of the blockchain.

3.4. Genesis Block: Each blockchain requires a starting block that does not reference a previous hash. This initial block is known as the "genesis block." It serves as the foundation of the chain and contains no values in the "previous hash" or "ballot data" fields.

3.5. Voter Verification Process: The voter verification process is designed to create a traceable audit trail for each vote, allowing voters to confirm their participation in the final tally. After an election, voters can verify their votes using the receipts provided at polling stations. By scanning a QR code or accessing the verification system online, voters are directed to their precinct's verification portal. The system confirms their vote was included in the tally without disclosing how they voted, ensuring privacy.

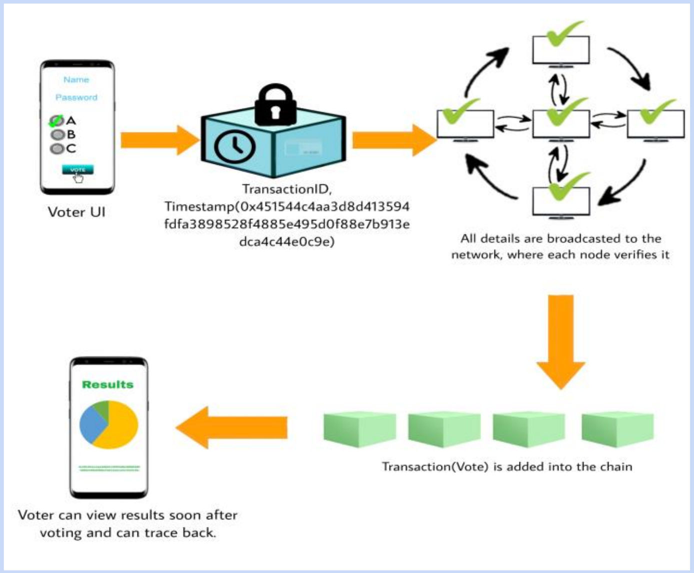

Name
Password
A
B
C
VOTE
Voter UI
TransactionID,
Timestamp(0x451544c4aa3d8d413594
fdfa3898528f4885e495d0f88e7b913e
dca4c44e0c9e)
All details are broadcasted to the network, where each node verifies it
Transaction(Vote) is added into the chain
Results
Voter can view results soon after voting and can trace back.

Figure 2. Voting process

3.6. Post-election Audit Process:RLAs verify election results using statistical methods, and this system supports three types of RLAs: ballot-level comparison, ballot pooling, and batch-level comparison.
Separate blockchains are maintained for each precinct, enabling the generation of individual tallies. To conduct a ballot-level comparison, ballots from a randomly selected precinct can be rescanned using different scanners and validators. This process creates a new chain, which can be compared to the original tally to verify accuracy.

CHAPTER 4

Implementation and Testing

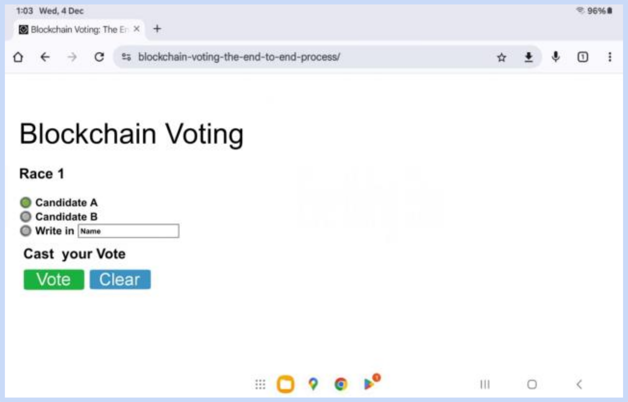
The development and evaluation of this study were carried out in two stages.

- Stage 1: A prototype was developed based on the voting algorithm discussed in Chapter 3.
- Stage 2: Three simulated elections were conducted to assess the performance of the system.

First Simulation: This phase focused on testing specific system features such as the zero-knowledge

4.1 Blockchain Voting Prototype: For the prototype, actual ES&S DS200 optical scanners were not used. Instead, software was developed to mimic the functionality of a ballot scanner, validator, and central server. A web form created with HTML and JavaScript was used to handle ballots. A web API (REST API) built with the Python Flask framework was employed to transmit ballot data to the validator. Each ballot was converted into a JSON object and stored within the block as "ballot data."

The validator was created using Python, with local chains stored as binary files within the validator.



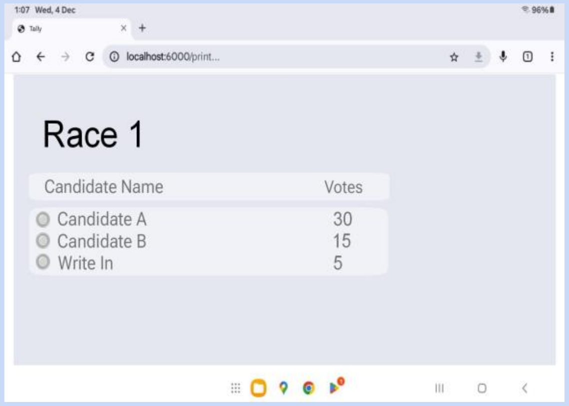
Blockchain Voting

Race 1

● Candidate A
○ Candidate B
○ Write in Name

Cast your Vote

Figure 3.Online Ballot form.

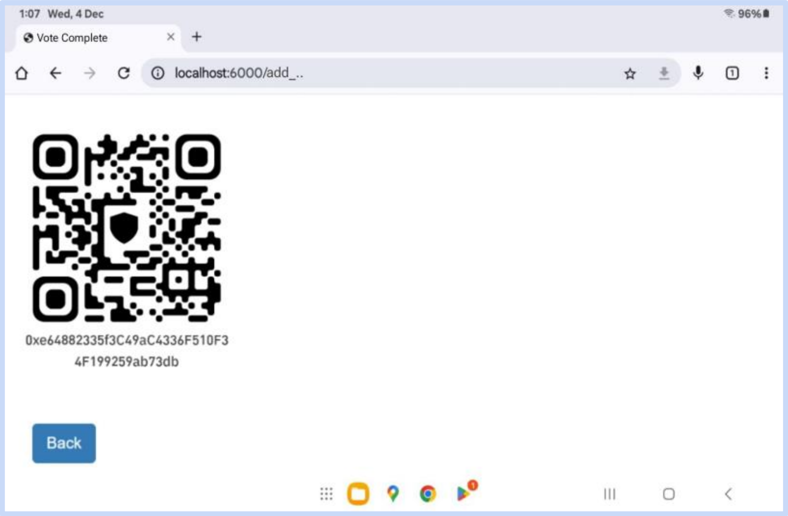


Race 1

Candidate Name	Votes
○ Candidate A	30
○ Candidate B	15
○ Write In	5

Figure 4.Tally API page

The central server's functionality was also implemented in Python on the same server as the validator. The validator authentication process (using the zero-knowledge protocol) and encryption of data exchanged between the central server and the validator were set up as outlined in Chapter 3. The proof of work log was stored as a binary file on the server. The API generates a webpage showing the current tally of votes. The voter verification process was slightly adjusted due to the absence of actual validators connected to a QR printer. Instead of printing a receipt with a QR code, a QR code image was shown on the online ballot page once a vote was successfully cast. Voters could download or print this image as a receipt to access the voter verification portal. As in the original design, the image could be scanned with a smartphone camera or a QR code scanner. The scanned image links to the voter verification portal, which is a web API built using the same Python Flask framework as the tally API. This API notifies the voter that their vote has been included in the final tally.Post-election audits were executed as Python scripts. These scripts access data stored in binary files to perform the necessary validations and ensure the election's accuracy. The voter verification and post-election audit processes are further detailed in the following sections.



0xe64882335f3C49aC4336F510F3
4F199259ab73db

Figure 5.Vote verification QR code.

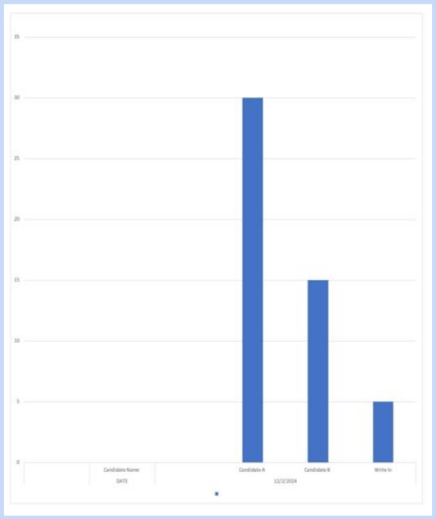


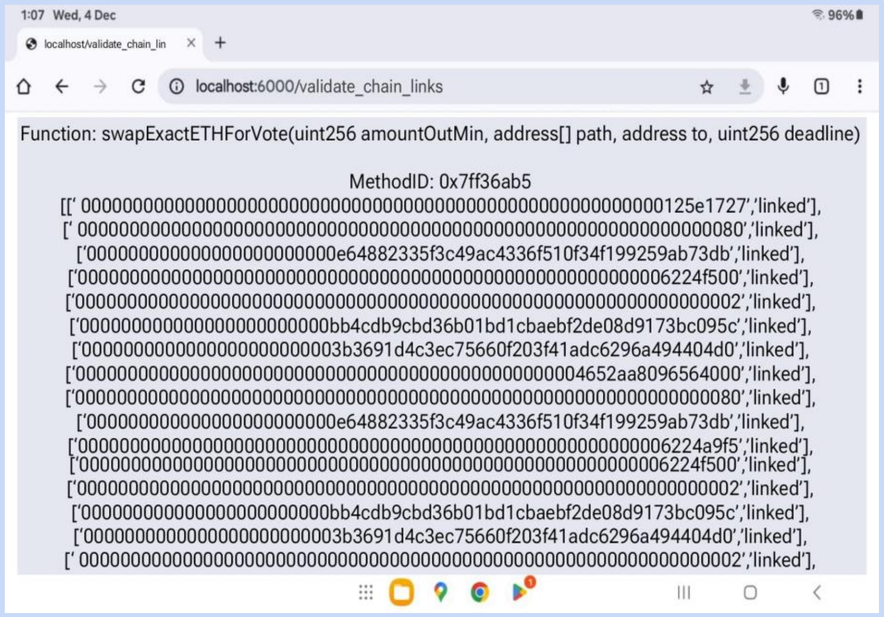
Figure 6: votes of candidates

4.2 First Trial Election process: As previously stated, the primary objective of the first trial election was to test the essential functions of the voting algorithm. Several votes were entered into the chain for each of the options to assess the success of the voting process.All keys and credentials needed for executing the zero-knowledge protocol and encrypting the data were manually generated before the trial and embedded as fixed values within the code. Comparing the generated hash and the corresponding POW for the block showed that hashes generated with an incorrect decryption key did not follow the correct POW pattern.

4.3 Second Trial Election process: The primary objective of the second mock election was to evaluate the computational complexity of the proposed proof of work algorithm. The mock election was designed to determine the average number of attempts required to find a correct hash based on the number of leading zeros

4.4 Third Trial Election process: The third mock election was held to mimic a real election setting. Two precincts were used, each with its own validator and vote chain. The use of two validators was intended to test the voter verification process and the batch-level comparison audit. Each precinct had slightly biased voting data. Instead of using the online ballot form, all votes were added directly into the system using the API Risk-limiting audits and other checks were conducted on each precinct's vote chain to assess the system's ability to perform post-election audits. The audit results and other election records were cross-verified against the tally API to validate the election outcomes.For the first precinct, simulated votes were cast, with 60% for Candidate A, 30% for Candidate B, and the remainder as write-ins (see Figure 4).

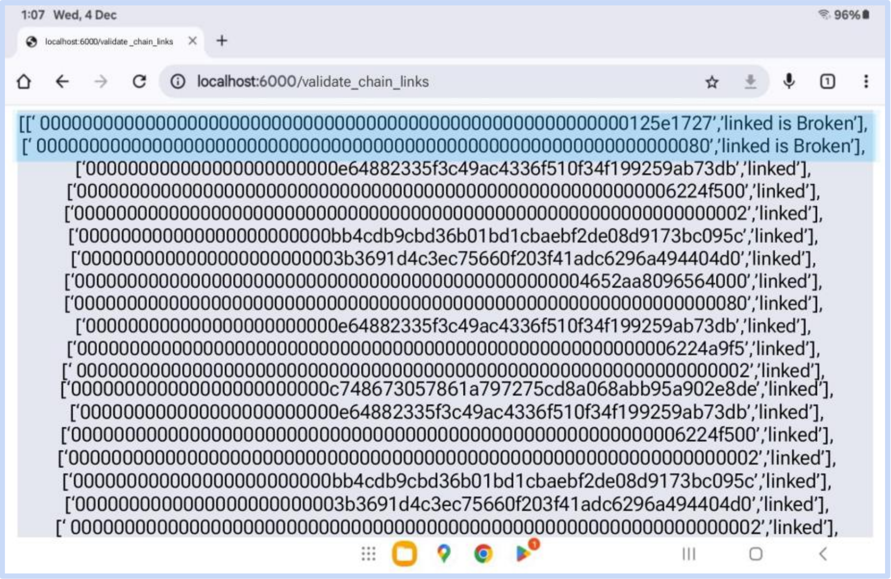
4.5.Voter Verification Process: The design of the QR code for verifying votes underwent modifications, leading to a change in its implementation. Instead of being printed on a paper receipt as originally outlined in Chapter 3, the QR code is now displayed on the screen after a ballot is successfully submitted online. The structure of the displayed QR code remains consistent with the format described (see figure.5)


1:07 Wed, 4 Dec
96%
localhost/validate_chain_lin
localhost:6000/validate_chain_links

```
Function: swapExactETHForVote(uint256 amountOutMin, address[] path, address to, uint256 deadline)

MethodID: 0x7ff36ab5
[[‘ 00000000000000000000000000000000000000000000000000000000125e1727’,’linked’],
[‘ 0000000000000000000000000000000000000000000000000000000000000080’,’linked’],
[‘000000000000000000000000e64882335f3c49ac4336f510f34f199259ab73db’,’linked’],
[‘000000000000000000000000000000000000000000000000000000006224f500’,’linked’],
[‘0000000000000000000000000000000000000000000000000000000000000002’,’linked’],
[‘000000000000000000000000bb4cdb9cbd36b01bd1cbaebf2de08d9173bc095c’,’linked’],
[‘0000000000000000000000003b3691d4c3ec75660f203f41adc6296a494404d0’,’linked’],
[‘0000000000000000000000000000000000000000000000004652aa8096564000’,’linked’],
[‘0000000000000000000000000000000000000000000000000000000000000080’,’linked’],
[‘000000000000000000000000e64882335f3c49ac4336f510f34f199259ab73db’,’linked’],
[‘000000000000000000000000000000000000000000000000000000006224a9f5’,’linked’],
[‘000000000000000000000000000000000000000000000000000000006224f500’,’linked’],
[‘0000000000000000000000000000000000000000000000000000000000000002’,’linked’],
[‘000000000000000000000000bb4cdb9cbd36b01bd1cbaebf2de08d9173bc095c’,’linked’],
[‘0000000000000000000000003b3691d4c3ec75660f203f41adc6296a494404d0’,’linked’],
[‘ 0000000000000000000000000000000000000000000000000000000000000002’,’linked’],
```

Figure 7. Block connectivity audit success



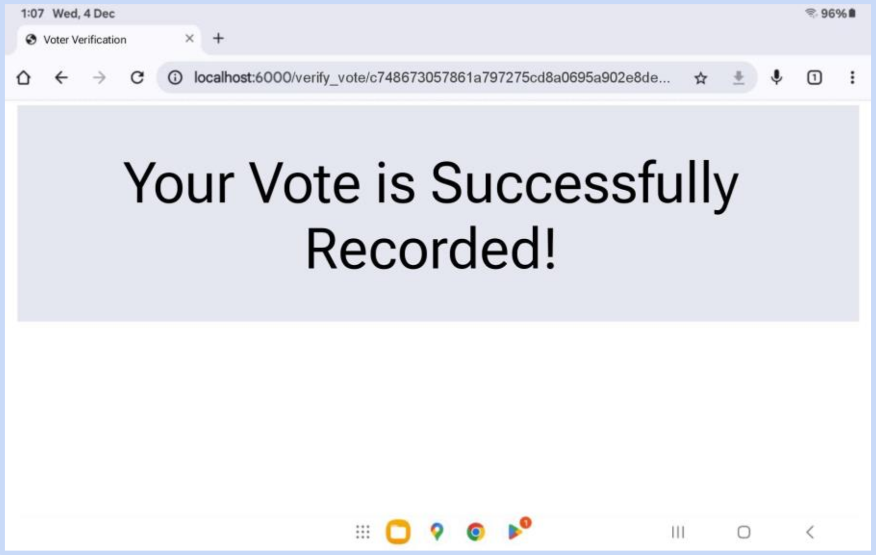
1:07 Wed, 4 Dec
96%
localhost:6000/validate _chain_links
localhost:6000/validate_chain_links

```
[[‘ 00000000000000000000000000000000000000000000000000000000125e1727’,’linked is Broken’],
[‘ 0000000000000000000000000000000000000000000000000000000000000080’,’linked is Broken’],
[‘000000000000000000000000e64882335f3c49ac4336f510f34f199259ab73db’,’linked’],
[‘000000000000000000000000000000000000000000000000000000006224f500’,’linked’],
[‘0000000000000000000000000000000000000000000000000000000000000002’,’linked’],
[‘000000000000000000000000bb4cdb9cbd36b01bd1cbaebf2de08d9173bc095c’,’linked’],
[‘0000000000000000000000003b3691d4c3ec75660f203f41adc6296a494404d0’,’linked’],
[‘0000000000000000000000000000000000000000000000004652aa8096564000’,’linked’],
[‘0000000000000000000000000000000000000000000000000000000000000080’,’linked’],
[‘000000000000000000000000e64882335f3c49ac4336f510f34f199259ab73db’,’linked’],
[‘000000000000000000000000000000000000000000000000000000006224a9f5’,’linked’],
[‘ 0000000000000000000000000000000000000000000000000000000000000002’,’linked’],
[‘000000000000000000000000c748673057861a797275cd8a068abb95a902e8de’,’linked’],
[‘000000000000000000000000e64882335f3c49ac4336f510f34f199259ab73db’,’linked’],
[‘000000000000000000000000000000000000000000000000000000006224f500’,’linked’],
[‘0000000000000000000000000000000000000000000000000000000000000002’,’linked’],
[‘000000000000000000000000bb4cdb9cbd36b01bd1cbaebf2de08d9173bc095c’,’linked’],
[‘0000000000000000000000003b3691d4c3ec75660f203f41adc6296a494404d0’,’linked’],
[‘ 0000000000000000000000000000000000000000000000000000000000000002’,’linked’],
```

Figure 8. Block connectivity audit failure

Each QR code provides a direct link to the vote verification portal and includes a hash of the four ballot data fields represented as a hexadecimal string.

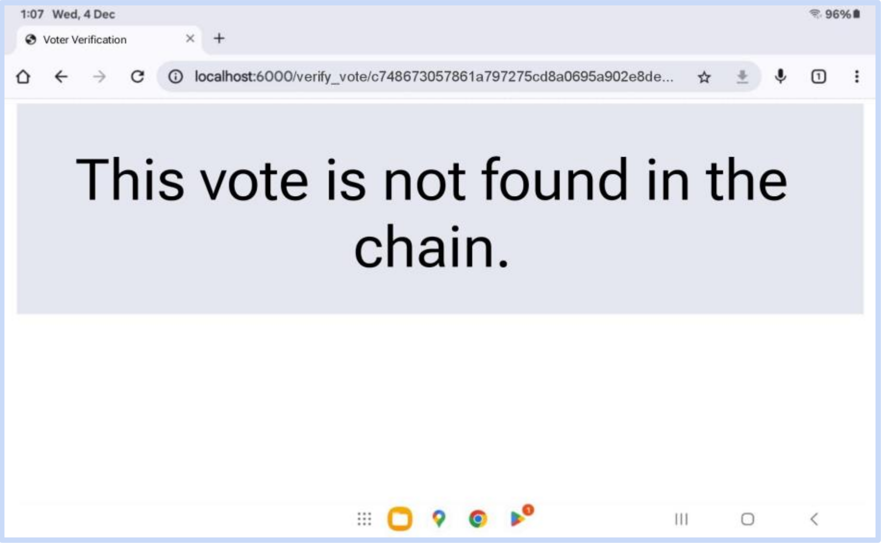
To evaluate the voter verification process, a generated QR code was scanned using a mobile device. The device automatically accessed the verification portal through the embedded link in the QR code. For legitimate votes, the system confirmed that the vote had been successfully recorded (refer to Figure 9). In cases of invalid votes, the system returned an error message (refer to Figure 10).



1:07 Wed, 4 Dec
96%
Voter Verification
localhost:6000/verify_vote/c748673057861a797275cd8a0695a902e8de...

Your Vote is Successfully Recorded!

Figure 9.Vote verification successfully completed


1:07 Wed, 4 Dec
96%
Voter Verification
localhost:6000/verify_vote/c748673057861a797275cd8a0695a902e8de...

This vote is not found in the chain.

Figure 10. Vote verification Failure

CHAPTER 5

Conclusion

The primary aim of this research is to leverage blockchain technology and the concept of proof of work to develop a voting system that is auditable, tamper-proof, and secure. The system also integrates two advanced security mechanisms in a novel manner: zero-knowledge protocols to prevent sharing sensitive data and end-to-end encryption via the El-Gamal public-key cryptosystem. It consistently authenticated validators with the central server, except for a few cases where the value of R (calculated as R= v-Cb) was negative. This issue was resolved by substituting gR in the equation U= gRBC mod n with the modular inverse of g-R when R was negative:

{If R>0:U= gRBC mod n}

{If : R<0:}

{U= mod inverse (g-R)BC mod n}

The El-Gamal cryptosystem was utilized to generate secret keys for the validators and to encrypt communication between devices. This approach successfully ensured secure, end-to-end encrypted communication between validators and the central server.

The efficiency and effectiveness of the proposed proof-of-work mechanism were evaluated during the second mock election. The experiment measured the average number of attempts and the average time required to generate a correctly formatted hash by incrementally increasing the sequence length of leading zeros in steps of two. The findings indicated that the average number of attempts grew approximately fourfold with each additional two zeros in the sequence. Similarly, the average time to generate a hash exhibited exponential growth as the sequence length increased, especially for longer sequences. This behavior is largely attributed to the computational time required for modular calculations within the zero-knowledge protocol.

Future Directions

Several potential enhancements have been identified for both the prototype system and the foundational algorithm. These could be explored further as part of future research.

Enhancements to the Prototype

A key enhancement for the prototype would involve conducting mock elections where multiple validators work concurrently to generate hashes. This would offer a more comprehensive evaluation of the system's performance.

Refinements to the Algorithm

One notable improvement to the algorithm would involve transitioning to a single blockchain shared among all validators within an election, rather than maintaining separate chains for each validator. This shift might necessitate the introduction of a consensus mechanism, similar to the protocol used in Bitcoin, to ensure blocks are securely added to the chain.

Another potential refinement would be implementing a private information retrieval protocol for the voter verification process. Such a protocol would allow voters to securely access information from the blockchain during vote verification without revealing the origin of the request. This enhancement would improve privacy by ensuring that no third party could trace the source of verification attempts.

List of References

1. "Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications"
Author: Imran Bashir
Edition: 3rd Edition
Publisher: Packt Publishing
Relevant Pages: Chapter 16 focuses on blockchain applications, including e-voting systems.
2. "Blockchain Basics: A Non-Technical Introduction in 25 Steps"
Author: Daniel Drescher
Publisher: Apress
Relevant Pages: Chapter 19 discusses the potential of blockchain for elections and voting systems.
3. "Blockchain Technology for IoT Applications"
Authors: Seok-Won Lee, Sabina Jeschke, Soon Hyeok An
Publisher: Springer
Relevant Pages: Section 7 covers case studies on voting systems using blockchain.
4. "Blockchain and the Law: The Rule of Code"
Authors: Primavera De Filippi, Aaron Wright
Publisher: Harvard University Press
Relevant Pages: Chapter 9 explores the legal implications of blockchain in governance and voting.
5. "Applications of Blockchain Technology in Business"
Editors: Mohsen Attaran, Sharmin Attaran
Publisher: Springer
Relevant Pages: Chapters on blockchain's role in voting and secure transactions.
6. "Blockchain Technology and Applications"
Authors: Pethuru Raj, Poornima M, Preetha Evangeline
Publisher: CRC Press
Relevant Pages: Chapter 8 details blockchain applications in e-voting systems.
7."Blockchain Technology and Applications"
Author: Kumar Saurabh, Ashutosh Saxena

Publisher: Wiley India
Overview: Covers fundamental concepts and advanced blockchain applications, including governance and voting systems.

8. "Blockchain for Beginners: Develop Your First App"
Author: Debjani Ghosh
Publisher: BPB Publications
Overview: Discusses blockchain basics and introduces practical applications, including secure voting systems.

9. "Blockchain Technology: Concepts and Applications"
Author: Lata Nautiyal, Rohit Tanwar, Pramod Kumar
Publisher: CRC Press (Indian edition available)
Overview: Includes use cases of blockchain in governance and voting systems.

10. "Blockchain Technology and Its Applications in IoT and Other Emerging Technologies"
Editors: R. Rajesh, Arvind Panwar, K. Ramesh
Publisher: Springer (Indian collaboration)
Overview: Explores blockchain's role in decentralized systems, with examples of its use in e-voting.