Decentralized Voting System Using Ethereum Blockchain

Kiran rathod[1], Sahana[2], Yashaswini [3],

[1]*kiran9008621624@gmail.com,* [2]*. sahanahn9886@gmail.com,* [3] *shivashivakumar3220@gmail.com,*
Under the Guidance of Dr. D. SIVAKUMAR, B.E., M.E., Ph.D.,

ABSTRACT

Voting is a cornerstone of democratic societies. However, traditional paper-based and electronic voting systems continue to encounter significant challenges, including tampering, lack of transparency, human error, and centralized control. Blockchain technology presents a secure alternative by leveraging decentralization, immutability, and distributed consensus. Ethereum further enhances these capabilities through smart contracts, which automate and enforce voting rules. This research introduces a decentralized voting system built on the Ethereum blockchain, utilizing smart contracts, cryptographic hashing, voter authentication, and a UTXO-based Vote Coin mechanism to guarantee one-voter-one-vote security and comprehensive verifiability. The system design is informed by the Voting Management System (VMS) framework proposed by Farooq et al. (2022), which incorporates layered security, chain validation, and resistance to common voting-related attacks.

Keywords: - Blockchain, Ethereum, Smart Contracts, Decentralized Voting, Distributed Ledger Technology, Cryptography, Consensus Algorithms, Election Security, Vote Verification.

1. Introduction

Voting is fundamental to democratic governance, enabling citizens to select leaders and influence public policy. Despite their importance, traditional paper-based and electronic voting systems remain susceptible to security vulnerabilities, lack of transparency, inefficiency, and high operational costs. These shortcomings can result in ballot tampering, vote suppression, miscounts, and logistical delays. While Electronic Voting Machines have improved counting speed, they are still exposed to risks such as hardware tampering, malicious firmware, insider threats, and centralized system failures. Furthermore, many digital voting platforms lack independent verifiability, which undermines voter trust.

Blockchain technology, introduced by Nakamoto in 2008, offers a decentralized and immutable ledger that removes the need for centralized authorities. Ethereum extends these benefits by enabling smart contracts that automatically enforce voting rules. Farooq et al. (2022) demonstrated that blockchain-based voting frameworks can address major electoral vulnerabilities by providing decentralized control, immutable records, voter authentication, real-time verifiability, and a layered architecture featuring UTXO-based Vote Coins and Chain Security Algorithms. These features position blockchain as a robust foundation for secure, transparent, and scalable national-level voting systems.

1.1 Problem Statement and Objectives

Traditional and electronic voting systems continue to face persistent issues, including tampering, human error, high costs, and dependence on centralized servers that are vulnerable to attacks or manipulation. Centralization also restricts transparency, preventing voters from verifying their own votes. In some regions, inadequate identity verification allows for duplicate or fraudulent voting.
This study aims to develop a decentralized, tamper-resistant, and transparent voting system using the Ethereum blockchain. The primary objectives are as follows:
Enforce one-voter-one-vote through smart contracts and Vote Coins

Secure votes using cryptography and consensus mechanisms
Enable real-time verification of votes
Detect tampering through a Chain Security Algorithm

Enhance usability via a decentralized application (Dapp) interface
Assess system performance and evaluate environmental and economic benefits of blockchain-based elections

2 System Architecture

The proposed system architecture consists of several integrated components designed to ensure a secure and transparent voting process. The Voter Interface (Dapp) allows users to register and cast votes, while the Authentication Authority verifies voter identity using MSISDN or National ID credentials. Core election operations are managed within the Smart Contract Layer, and all validated transactions are recorded on the Ethereum-based Blockchain Network to guarantee immutability and decentralization. The Security and Consensus Layer provides cryptographic protection and block validation, safeguarding the system against tampering and unauthorized modifications. Additionally, the Verification and Auditing Module enables both voters and officials to confirm the integrity and accuracy of votes. Collectively, these components establish a reliable end-to-end voting architecture.

Voter Interface (Dapp)
Authentication Authority
Smart Contract Layer
Blockchain Network (Ethereum)
Security and Consensus Layer
Verification

These elements interact to deliver a decentralized, transparent, and auditable voting system.

2.1 High-Level Architectural Flow

The system's high-level flow is as follows:- User (Voter): Accesses the Voter Dapp UI (mobile or web application)
Authentication Authority: Verifies voter credentials
Smart Contract Layer (Ethereum):
Registration Contract
Voting Contract
Vote Coin (VC) Mechanism
Blockchain Network (Nodes):
Block Creation
Transaction Storage
Distributed Ledger
Public Verification Portal: Enables independent verification of votes
System Layers and Key Components

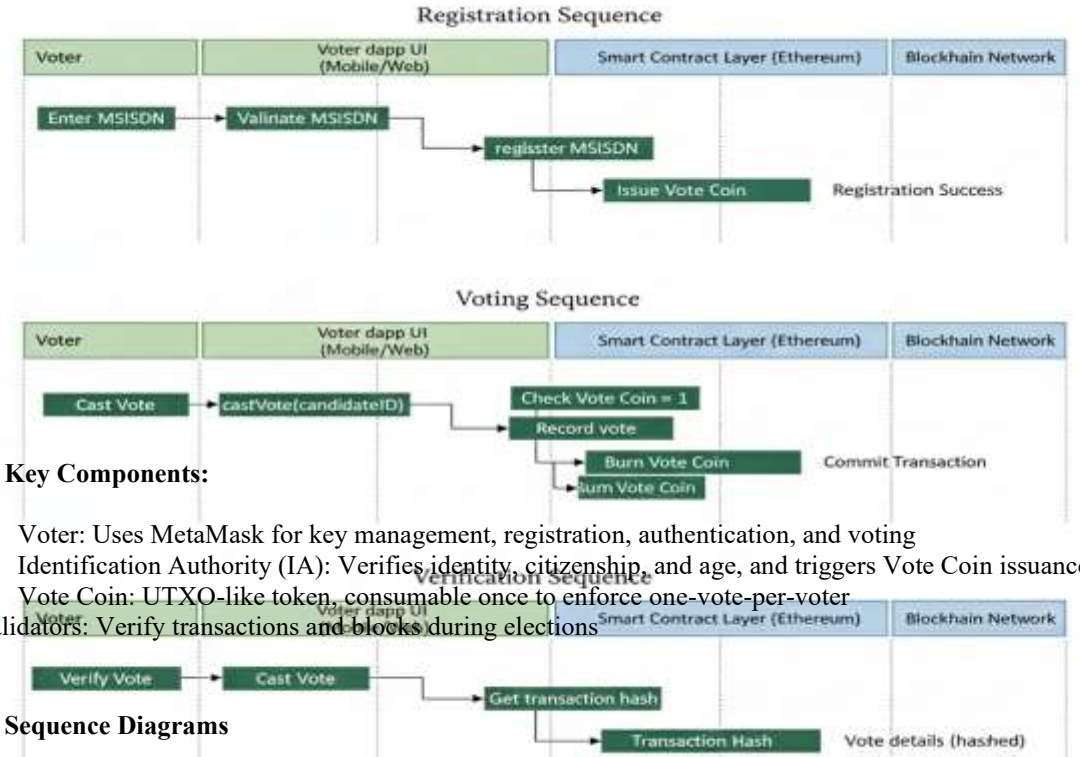
Interface Layer: Voters interact with the Dapp via web browsers using Web3.js. The interface supports multiple languages, straightforward navigation, secure MetaMask login, candidate selection, vote submission with confirmation, and vote verification through transaction hashes.

Application Layer: Manages pre-voting logic, including MSISDN/National ID verification, registration forwarding, eligibility checks, and vote status retrieval. It connects trusted data sources to smart contracts for secure voter identification.

Trust Layer (Smart Contracts): Immutable contracts enforce voting rules. The Registration Contract issues one Vote Coin per unique voter, while the Voting Contract validates and burns the Vote Coin upon voting and records the vote. An optional Administrative Contract manages candidates and results.

Blockchain Layer: Stores votes as hashed transactions in distributed blocks, ensuring replication and verification for data integrity.

Security Layer: Utilizes hashing, digital signatures, nonces, block linking, and consensus algorithms. A custom Chain Security Algorithm validates proofs and rejects malicious blocks or transactions.



2.2 Key Components:

Voter: Uses MetaMask for key management, registration, authentication, and voting

Identification Authority (IA): Verifies identity, citizenship, and age, and triggers Vote Coin issuance

Vote Coin: UTXO-like token, consumable once to enforce one-vote-per-voter

Validators: Verify transactions and blocks during elections

2.3 Sequence Diagrams

Registration: Voter submits ID to Dapp → Application Layer → Identification Authority verifies → Signals Registration Contract → Issues Vote Coin

Voting: Voter selects candidate → Dapp → Voting Contract validates Vote Coin → Burns Vote Coin, stores vote hash

Verification: Voter inputs transaction hash → Dapp queries Blockchain Layer → Displays immutable vote status

3. Methodology

The methodology for designing and developing the decentralized voting system on the Ethereum blockchain follows structured system engineering practices. The process includes requirement analysis, system modelling, smart contract development, frontend, and backend implementation, blockchain setup, and comprehensive testing.

Functional requirements focus on secure voter registration and authentication, enforcement of one-voter-one-vote through a Vote Coin mechanism, immutable on-chain vote storage, automated tallying, and real-time verification via transaction hashes. Non-functional requirements address security, transparency, scalability, usability, low latency, and immutability.

System modelling identifies key actors—voters, the Identification Authority, smart contracts, blockchain nodes, and administrators—and maps their interactions to ensure a seamless voting workflow.

4. Implementation and results

The decentralized voting system was implemented using contemporary blockchain technologies, secure

authentication processes, and user-friendly interfaces. Solidity was used for smart contract development, MetaMask for wallet and transaction management, and Ganache for simulating a local blockchain network. Backend services were developed with Next.js, and Web3.js facilitated communication between the application and the Ethereum blockchain. The frontend was implemented using React.js with Node.js support, ensuring a responsive user experience.

Core functionalities include:

Voter Registration

```
{
   function register(address _voter, uint _msisdN) {
   function register(address _voter, uint _msisdN) public {
      require(!registred[_voter], "Already registred");
      registred[_voter] = true;
      voteCoin[_voter] = 1;   // Issue Vote Coin
   }
 }
```

Listing 1. Voter Registration function showing duplicate check and vote coin issuance.

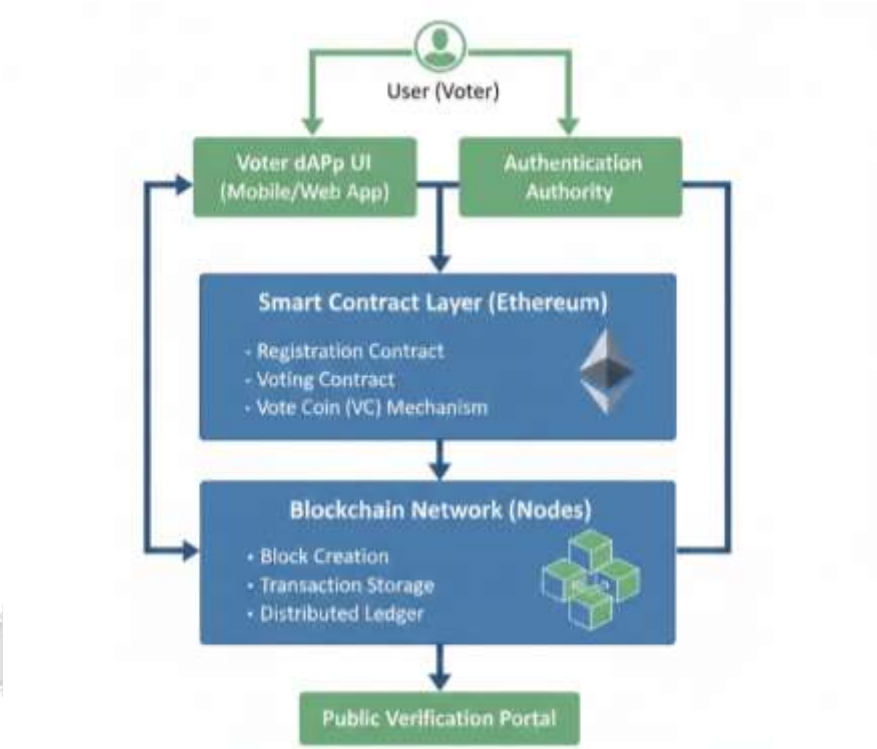
Vote Casting Logic

```
{
   function cast(uint _candidateID) public {
      require(registred[msg.sender], "Not registered");
      require(voteCoin[msg.sender] == 1, "Vote already cast");
      votes[_candidateID]++;
      voteCoin[msg.sender] = 0;  // Consume Vote Coin
      emit VoteCast(msg.sender, _candidateID, block.timestamp);
   }
}
```

Listing 2. Vote Casting function with registration check, single vote enforcement, vote tallying, and event emission.

Each vote generates a unique transaction hash, which voters can verify on Ethers can, ensuring transparency and independent confirmation. Testing results indicate that response time increases proportionally with transaction load, with minimal delay for approximately 500 voters and moderate congestion beyond 1,000 voters. Chain size evaluations show that the blockchain grows linearly as new blocks are added, demonstrating efficient storage management and scalability for large-scale voting.

4.1 high -level Architectural



5. Advanced Features and Innovations

The decentralized voting system incorporates several advanced features to address the limitations of traditional and electronic voting systems. These enhancements improve security, verifiability, scalability, and reliability. Many of these innovations are based on the transparent blockchain voting framework by Farooq et al. (2022), which introduced the UTXO voting model, Chain Security Algorithm, and mechanisms resistant to attacks at the national scale.

5.1 Chain Security Algorithm

The Chain Security Algorithm (CSA) safeguards the blockchain by verifying each block's previous hash, validating the proof mechanism, ensuring all transactions are legitimate and non-duplicated, and detecting any malicious alternative chains. This process maintains the integrity and reliability of the voting system.

5.2 Additional security and verification features include:

Restricting block validation to trusted validators during elections to prevent 51% attacks

Utilizing Proof of Authority or hybrid consensus models, continuous hash power monitoring, and the CSA to detect inconsistencies. Applying cryptographic techniques such as SHA-256 hashing, public–private key encryption, and digital signatures to secure vote data and protect voter privacy. Supporting flexible consensus mechanisms (PoW, PoS, PoA, Ripple, Proof of Trust) for scalability and performance optimization. Providing each voter with a transaction hash, candidate identifier, and timestamp for independent vote confirmation

through blockchain explorers. Implementing decentralized identity verification using MSISDN or National ID to ensure voter uniqueness and prevent impersonation

5.3 Environmental Impact

Blockchain-based voting systems offer considerable environmental benefits compared to traditional paper-based and electronic voting methods. By eliminating the need for printed ballots, envelopes, and physical records, these systems can reduce paper consumption by up to 95%, thereby lowering deforestation, manufacturing pollution, and waste generation. The removal of ballot material transportation and storage further decreases CO_2 emissions, fuel usage, and logistical costs.

With Ethereum's transition to Proof of Stake, energy consumption has decreased by over 99.5%, making it a highly efficient platform for digital elections. Additionally, blockchain systems reduce hardware waste associated with Electronic Voting Machines by relying on existing personal devices and distributed servers, thus minimizing e-waste and environmental impact.

6. Conclusion

This research demonstrates that a decentralized voting system based on the Ethereum blockchain can effectively address major challenges in traditional and electronic voting. The system ensures transparency, immutability, voter uniqueness, and real-time verification. The multi-layer architecture, inspired by Farooq et al. (2022), combined with the Chain Security Algorithm and consensus mechanisms, offers robust protection against tampering and fraud. Implementation results confirm reliable performance, successful vote verification, and clear environmental and economic advantages. Overall, blockchain-based voting presents a secure and efficient alternative for modern elections and holds significant potential for future adoption in digital democratic systems.

7. Acknowledgement

The authors extend their sincere gratitude to the faculty members and project mentors for their ongoing guidance and valuable technical insights, which have greatly enhanced the quality of this work. Appreciation is also due to the open-source blockchain communities, Ethereum developers, and researchers whose tools and documentation supported the development of this decentralized voting system. Special thanks are given to Farooq et al. (2022) for their comprehensive blockchain voting framework, which significantly influenced the system's architectural and security design. The authors also thank peers, colleagues, and friends who participated in testing, provided feedback, and offered encouragement throughout the development process.

8. Result

Latency Analysis

Latency represents the total time required for a vote transaction to be confirmed and recorded on the Ethereum blockchain.

Number of Voters	Average Latency (seconds)
100	2–4
500	4–6
1000	6–8

9. References

[1] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[2] G. Wood, Ethereum: A Secure Decentralized Generalized Transaction Ledger, Ethereum Foundation, 2014.
[3] J. Bonneau et al., "Research Perspectives and Challenges for Bitcoin and Cryptocurrencies," IEEE Security & Privacy, vol. 14, no. 2, pp. 104–120, 2015.
[4] M. Swan, Blockchain: Blueprint for a New Economy, O'Reilly Media, 2015.
[5] M. Pilkington, "Blockchain Technology: Principles and Applications," Research Handbook on Digital Transformations, pp. 225–253, 2016.
[6] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," IEEE SPW, 2015.
[7] G. Noizat, "Blockchain Electronic Voting: A Systematic Review," Future Internet, vol. 10, no. 4, p. 75, 2015.
[8] Y. Wang, A. Kogan, and K. Kogan, "Designing Blockchain-Based Electronic Voting Systems," Information Systems, vol. 78, pp. 32–45, 2018.
[9] X. Xu et al., "The Blockchain as a Software Connector," IEEE International Conference on Software Architecture, 2017.
[10] M. Risius and K. Spohrer, "A Blockchain Research Framework," Business & Information Systems Engineering, 2017.
[11] R. Beck et al., "Governance in the Blockchain Economy," Journal of the Association for Information Systems, 2018.
[12] H. M. Kim and M. Laskowski, "A Perspective on Blockchain Smart Contracts," IEEE Internet Computing, 2018.
[13] C. Cachin, "Architecture of the Blockchain System," Cryptology ePrint Archive, 2016.
[14] K. Patel, "Security Analysis of Blockchain-Based Voting Systems," Journal of Computer Science, 2020.
[15] N. Szabo, "Formalizing and Securing Relationships on Public Networks," First Monday, 1997.
[16] A. Antonopoulos and G. Wood, Mastering Ethereum, O'Reilly Media, 2017.
[17] J. Dlugosch and N. Josuttis, "Testing Smart Contracts," Blockchain for Enterprise, O'Reilly, 2018.
[18] M. Dworkin, "SHA-256 Cryptographic Hash Algorithm," NIST, 2015.
[19] A. Menezes et al., Handbook of Applied Cryptography, CRC Press, 1996.
[20] M. S. Farooq, U. Iftikhar, and A. Khelifi, A Framework to Make Voting System Transparent Using Blockchain Technology, 2022.