International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Secure Voting System Using Blockchain Technology: A Decentralized Approach to Enhance Electoral Integrity

Suraj Yadav, Sagar Gupta, Tanishq Raj Mahaur, Mr. Anurag Anand Duvey
Computer Science (Artificial Intelligence), Poornima Institute of Engineering & Technology (Jaipur).

Abstract- The conventional electronic voting machines often have problems like, centralized vulnerabilities, lack of transparency, prone to single-point-of-failure attacks, as well as high administrative overhead for voter verification. This paper presents a solution: A Decentralized e-voting framework created on the Ethereum Virtual Machine (EVM) that tackles these issues through the integration of blockchain immutability and mobile-native biometric authentication. With the medium this project, we propose a system that implements Solidity smart contracts to manage election lifecycles and a React Native frontend for User friendly UI and cross-platform accessibility. The key innovations such as a cycle-based state management mechanism for optimizing the contract reusability and a cryptographic credential-hashing protocol that safeguards voter identity without the need for high-cost third-party verification services.

Keywords- Blockchain, Electronic Voting System, Smart Contract, Ethereum, Decentralized Ledger, Cryptography, Secure Voting, Transparency, Immutability, Authentication, Aadhar Verification, OTP Validation, Mobile Voting App, Edge Computing, Distributed System, Election Security, Data Integrity, Air-Gapped Hardware, Trusted Setup, Digital Democracy.

I. INTRODUCTION

The integrity of the democratic election relies on the security and fairness of the voting system. Conventional voting systems, from paper ballots to centralized Electronic Voting Machines (EVMs), are increasingly condemned for their lack of verifiability and vulnerability to Sabotaging.

As for now, due to its important characteristics like immutability, transparency, and decentralization, Blockchain technology has developed as a transformative resolution for e-voting by using Smart Contracts. Smart Contracts are self-executing code stored securely on the blockchain. It is possible to hardcode and implement election rules without human assistance. Although, several recent blockchain voting systems face issues like practical implementation, essentially regarding the high cost of voter authentication (e.g., SMS-based OTPs) and the complexity of handling voter's identities while maintaining confidentiality.

Problem Statement

The Identification Gap remains a critical hurdle despite blockchain provides a secure logbook. Verification of voters' registration often based on government centralized databases or expensive APIs of third-party providers, which causes limitations in cost and scalability.

Additionally, many decentralized systems don't have the user-friendly interface required for mass implantation on diverse mobile hardware.

With this paper, A proposal for a secure, end-to-end encrypted decentralized voting application, which developed using the Hardhat environment and the React Native framework for cross platform computability. This project is the implementation that bridges the gap between high- level blockchain security and practical usage. The primary contributions efforts of this work are as follows:

- **Biometric Integration:** This project implements a verification mechanism that utilizes mobile-native biometric hardware (such as FaceID/Fingerprint) to authenticate users locally, as well as mapping these identities to cryptographic hashes on the EVM to ensure zero-knowledge identity verification.
- **Optimized Smart Contract Architecture:** Introduction to a nested mapping and loop- based state machine in Solidity that allows for the recycling of the deployed

contract infrastructure. Which allows multiple election cycles without damaging historical data integrity.

- **Cross-Platform Decentralized UI:** This project also demonstrates a mobile GUI capable of interacting with a local Ethereum node over a Local Area Network (LAN), providing a blueprint for scalable, hardware-independent voting booths.

II. LITERATURE REVIEW

The literature exploration of electronic voting lasts several decades, with an increasing phase from centralized digital models to decentralized cryptographic systems.

A. **Vulnerabilities in Traditional and Centralized Electronic Voting:** Although offering high physical transparency, the traditional paper-based systems are badly known for prone to human errors, ballot stuffing, manual manipulation, and extended transportation logistical delays during the counting phase. The transformation from traditional ballot counting to Electronic Voting Machines (EVMs) rapidly improved processing speed and reduced invalid votes. However, EVMs introduced new difficulties concerning data security, software glitches, and proneness to hacking, this is because of security measures are not able to keep up the pace with sophisticated hackers. With the help of Standard online voting systems further expanded voter participation by allowing remote access for voting, however it also introduced several risks, like probality to DoS attacks, malware, identity theft, and centralized server hacking, hence failing from lack of verifiability.

B. **The Sudden from Shift to Blockchain-Based E-Voting:** The development of blockchain technology has brought a drastic change in the design of electoral systems. Blockchain works on the concept of decentralization i.e not having a central control, immutability i.e No changes in data once recorded, and transparency to publicly verify transactions.

C. Blockchain for Election Integrity and Transparency

The Blockchain technology assists as a distributed, immutable record that reduces the risks tied with the use of centralized database management. The research shown by Ahmed et al. [2] implies that conventional e-voting systems lack transparency and are prone to central server attacks, EtherVote [4] proposed a non-central, serverless e-voting model depending only on Ethereum smart contracts, illustrating that removing standard database layer significantly optimizes the system encryption and integrity. Furthermore, systematic reviews of the field between 2022 and 2025 [3], [5] identify blockchain as the primary catalyst for "universal verifiability," allowing voters to independently audit the election results without compromising secrecy.

D. **Cryptographic Advances and Authentication Models** Authentication and the preservation of anonymity represent the most complex dichotomy in e-voting design. According to the systematic review by Sánchez et al., modern blockchain voting protocols heavily rely on advanced cryptographic mechanisms such as Zero-Knowledge Proofs (ZKPs), homomorphic encryption, and blind signatures to obscure the vote's content while ensuring the voter's legitimacy. ZKPs allow a system to verify the validity of a vote without exposing the actual choice, thus preserving anonymity. Furthermore, research highly emphasizes multi- factor authentication. Studies have proposed integrating Aadhaar (India's biometric ID system) or other national identity databases with blockchain, where the physical biometric data or virtual ID is converted into a digital signature. Integrating these biometric features with Hyperledger Fabric or Ethereum ensures that only legitimately registered voters can cast a ballot, entirely neutralizing double-voting and impersonation threats.

E. Biometric Authentication in Digital Identity

A critical challenge in decentralized voting is the "Identity Gap"—ensuring that a cryptographic wallet address corresponds to a unique, eligible human voter. Kumar et al. [1] emphasize that biometric identification is resistant to identity theft and "double spending" of votes. Integrating biometrics provides a layer of physical security that passwords cannot replicate. Recent studies [6] suggest that mobile-native biometric hardware (FaceID/Fingerprint) offers a low-friction method for voter authentication, though researchers note the ethical and technical challenges of storing sensitive biometric templates on-chain.

F. EVM Optimization and Scalability

For Blockchain based voting system to be used in national election, the scalability remains a significant hurdle. Many structures suggested hybrid blockchain systems to balance public transparency with private data [1]. Inventions like the

AdapT v2.0 design pattern [4] focuses on optimizing operating memory of blockchain EVM by lowering unnecessary duplication of smart contract. With that in mind, our proposed solution associates with these optimization trends by granting contract recyclability by reusing the contract, effectively handling the storage of voter logs across multiple cycles securely within a single deployed contract case.

Research Gaps and Identified Challenges

The prior works on blockchain indicates several critical research gaps. The main challenge for national scale elections remains scalability of blockchain system, because, large-scale elections requires public blockchains like Ethereum have problems with high transaction throughput and gas fees. Additionally, having balance in voter privacy with systemic transparency, handling the high computational expenses of advanced security, and navigating regulatory acceptance continue to hamper real-world deployment. Several designed models remains in prototype phases as well as lacking tough testing in highly questioned, large-scale public elections.

III. PROPOSED METHODOLOGY

For successfully designing and deployment of the Secure Voting System using Blockchain, a highly structured, sequence methodology has been used. The project strictly follows the Agile Project Management methodology, specifically utilizing the Scrum framework by drafting the needs in Product Backlog and dividing the steps to achive those needs in Spirint Backlogs which is very helpful to blockchain development due to the evolving nature of smart contract security, which often requires iterative testing.

Agile Project Management and Scrum Execution The development lifecycle of this project was divided into Sprints to make sure of continuous feedback, improvement of security auditing, and feature additions.

- **Sprint 1:** Core Functionality: The first sprint focused on implementing the basic architecture of the Voting Mobile App. The main aim was to deliver a working demo of the app, with all working functions of the blockchain voting system. It consists of secure user registration, identity recording, as well as basic vote casting. Within this backlog, the basic Solidity smart contracts (Voting.sol) were created and deployed on a local Ganache blockchain sever. The frontend UI was connected to the blockchain, allowing a users register, sign-in and allow transaction through their wallet, which was then synchronously written to the contract state.
- **Sprint 2:** System Hardening and Off-Chain Scaling: After creating the functional core, Sprint 2 focuses on business-scale improvement and User Experience (UX) enhancements. IPFS was added to handle large media content, converting the system into an business-ready platform. Moreover, an off-chain analytics logic using JavaScript was created to analyze real-time voter percentages and visualize data without executing expensive on-chain computations.
- **Sprint 3:** Improved Security and Role-Based Access Control: This backlog emphasis on implementing accurate role-based access controls (RBAC) for Candidates, Voters, Administrators, and Auditors. The Soliditery logic was improved to explicitly prevent double voting and secure the vote validation processes to improve the security. To improve more privacy, encryption systems were heavily integrated, allowing vote receipts to be generated securely, and transparency was optimized .
- **Sprint 4:** Testing, Auditing, and Final Deployment: This phase of sprint was dedicated entirely to the testing of system and reviewing vulnerabilities. A Complete functional, encryption, and working of system testing practices were executed to simulate high density, to confirm reliability of the system.

System Architecture

The project tries to implement a decentralized three-level architecture, which consists of: the Client Layer (Frontend/GUI), the Communication Layer (JSON-RPC over LAN), and the Blockchain Layer (Solitary Smart Contract).

A. Smart Contract Design and State Management

The core concept of the working system is the EVMVotingDemo smart contract, written in Solidity. This contract functions as a deterministic state machine rather than traditional databases. We uses an enum to accurately enforce election phases:

1. **NotStarted:** The system is open for voter registration, but the castVote function as well as election is logically locked.
2. **Active:** The Admin activates the transition to enable this state, allowig the voting mechanism to startup.

3. **Ended:** Voting shut-downs, and the contract further retrieves the final tallies.

Here, The Nested Mapping is the key innovation in our system architeure that is been used for storage, allowing the contract to ensure an electionCycle variable. By increasing this variable one-by-one this variable during a reset, the system effectively creates a new and "fresh" registry for each election cycle, while also maintaining the historical integrity of previous cycles.

B. Cryptographic Identity Mapping

To combine the physical identity with blockchain encryption, we implement a two-factor verification mechanism:

- **Local Biometric Verification:** The React Native client uses the expo-local- authentication library to start the users' device hardware scanner, whether it's a Fingerprint or FaceID right before any transaction is initiated. This allows the "Physical Presence" of the voter without transmitting biometric data to the blockchain network.
- **Credential Hashing:** Upon successful biometric scan, the system concatenates the user's unique identifier (Mobile Number) and a private salt (Password). This string is processed through the Keccak-256 hashing algorithm via the ethers.id() function.

C. Network Topology and Connectivity

The entire model was running on a local Ethereum network using the Hardhat development environment. To simplify cross-device accessibility which was a requirement for a multi-voter system, The Hardhat node is configured to take input from on the wildcard host: npx hardhat node --hostname 0.0.0.0 This allows the React Native frontend to communicate with the Voting System over a Local Area Network (LAN) via the JSON-RPC mechanism. The frontenf acts as a lite provider. communicating with the deployed contract address using a pre-compiled Application Binary Interface (ABI).

D. Data Integrity and Preventative Measures

The systems implements several failsafes in the smart contract logic to ensure smooth and accurate usage:

- **Double-Voting Prevention:** The Voter struct consists of a hasVoted Boolean to avoid users to vote who already voted. Before a vote is logged, the contract rechecks this flag for the current electionCycle.
- **Administrative Access Control:** The transitions of the Finite state machine (Start/End/Reset) are secured by a require(msg.sender == admin) function, which makes sure that only the authorized admin can manage the election lifecycle.

IV. IMPLEMENTATION AND ALGORITHMS

The implementation of the system architectural design to relate with the distinct programmable logic. The basic operational stream is segmented to Registration, Authentication, Voting, and Counting levels.

Database Design and Smart Contract State

For maintaining desired efficiency on the Ethereum blockchain network, the state values are accurately defined -

- **Voter Struct:** struct Voter { string name; string aadhaar; string mobile; string password; bool hasVoted; }. This is the defined structure of the links of voter's real-world credentials and data to their distinct blockchain state, specifically using the hasVoted boolean to avoid duplicate transactions.
- **Candidate Struct:** struct Candidate { uint id; string name; string party; string ipfsHash; uint voteCount; bool isApproved; }. This is the candidate's profile and manifesto which consists of media like posters and video are available through the ipfsHash, while voteCount acts as the decentralized storage.
- **State Mapping:** mapping(address => Voter) public voters; establishes the cryptographic linkage between the voter's public MetaMask address and their registered data.

A. Development Environment and Software Stack

The initial system was established using a full-stack JavaScript/TypeScript programming language to ensure instant increments and cross-platform as well as cross device compatibility.

- **Blockchain Backend:** A local Ethereum network was development, and the environment was established by utilizing Hardhat v2.22. The smart contracts were created in Solidity v0.8.24 ensuring compatibility with modern gas optimization features for better effectiveness.
- **Mobile Frontend:** The UI was developed by using React Native through the Expo SDK packages. This is a versatile choice that facilitated the integration of native device hardware for the biometric scanner, while granting a responsive web-based fallback for testing.
- **Middleware:** Ethers.js v6.x: This served as the main supporting backbone through its libraries for

communicating with the blockchain. It managed establishing connections, wallet and transaction management, as well as encoding/decoding of contract ABIs.

Smart Contract Implementation

The deployment of Solidatry Smart Contract was programmed by utilizing Hardhat Ignition, which make sures the deterministic and reusable contract addresses across multiple development environments. For the time begin and for testing purposes, the EVMVotingDemo contract was deployed with a hardcoded set of seven candidates to simulate a standard ballot.

Key implementation of this logic included:

- **State Control:** The resetElection function allows the administrator to clear vote tallies and increment the electionCycle without redeploying the contract, as seen in the provided source code.
- **Relayer Pattern:** In order to make the user experience simple for the demo, the system applied a hardcoded admin private key to give gas fees for voter transactions i.e Registration and Voting, mocking a "Gasless Transaction" environment for the end- users.

Biometric and Identity Integration

The implementation of the biometric identification and its integration was done by handleBiometricAuth function in the register.tsx and login.tsx files successfully aided by the expo-local-authentication API. It was done in the following ways-

1. **Hardware Check:** The system first launches a queries hasHardwareAsync() to verify the presence of a scanner as such fingerprint scanner or face scanner.
2. **Enrolment Check:** The system verifies if the user has a fingerprint or FaceID registered via isEnrolledAsync(). In their device and validates the registration.
3. **Authentication:** After successful hardware verification, the app advances to create the Keccak-256 identity hash, which makes sure that biometrics act as a gatekeeper for blockchain writes for secure ‘On person’ registration.

Experimental Results and User Interface

This entire prototype system was tested and verified using multiple physical devices connected over a Local Area Network (LAN).

- **Transaction Latency:** Starting from the local Hardhat node, the average time (in ms) from clicking "Vote" button to receiving the blockchain confirmation and verification was consistently under 200 milliseconds. The latency was very low for keeping user confidence in digital voting systems.
- **UI/UX Evaluation:** To improve readability while also maintaining the modern asthenic look for appealing mobile app, the interface was modernized to by enforcing a high-contrast light theme.
- **Security Validation:** The prototype also verified that the hasVoted function had successfully prevented constant voting attempts within the same cycle to avoid duplicate voting. Moreover, unauthorized tries to access the Admin Dashboard from a non-admin wallet were effectively denied by the smart contract's onlyAdmin modifier.

Security Analysis and System Resilience

- **51% Attack:** More than half of the attacks occurs when a malicious individual gains control over the majority of the blockchain network's hashing power or staked assets, supposedly, allowing them to revise the transaction history of the system. Nevertheless, when deploying the model on a robust and vast public blockchain such as Ethereum, which uses a large, globally decentralized Proof-of-Stake (PoS) agreement mechanism, executing a 51% attack is practically impossible as well as extremely expensive.
- **Distributed Denial of Service (DDoS):** DDoS attacks are the most commonly used attacks against traditional centralized servers are highly susceptible to vulnerabilities like flooding the system with traffic, causing the voting portal inaccessible. The decentralized nature of blockchain allows to counter this. The Ethereum network is reinforced by hundreds and thousands of distributed nodes; if some particular nodes are targeted and caused them to go offline, the broader network continues to process and validate voting transactions uninterrupted.
- **Sybil Attacks:** During a Sybil attack, an malicious entity creates very large amounts of fake identities to cast multiple votes and sabotage the election outcome. The proposed system brings a solution to this by entirely neutralizes this threat through its rigorous Multi-Factor Authentication mechanisms. This is because verifiable physical identity such as e.g., Aadhaar are encrypted using cryptographically bounding of each allowed MetaMask address, which requires a real-time OTP sent to a verified

mobile device. Henceforth, an attacker cannot programmatically mass-generate voting identities.

V. CONCLUSION

This paper presents a strong, decentralized e-voting system that successfully addresses critical challenges such as transparency, identity verification, and administrative overhead and problems like cyber attacks, identity theft, and other malicious practices. By improving the Ethereum Virtual Machine for immutable record handling and mobile-native biometric verification using local hardware for valid authentication, the developed system establishes a secure environment where election integrity is mathematically guaranteed.

The implementation of innovations like of a cycle-based state machine and nested mapping for identified users allowing for sustainable, reusable smart contracts architecture that minimizes deployment costs. Also, the cryptographic hashing of user credentials improves encryption as well as makes sure that voter privacy is maintained while preventing multiple-voting from single user and unauthorized access to admin controls.

Practical demonstration also indicate the results that the model is not only highly secure but also grants effectiveness such as the low-latency and high-accessibility which are required for modern democratic applications. This project shows that a high-encryption of secured blockchain is the solution that can be delivered with a user-friendly, cross-platform interface suitable for diverse voter demographics and election practices.

REFERENCES

1. N. Kumar, "Modernizing Voting Systems: A Comprehensive Approach Using Blockchain, Biometrics and Zero Knowledge Proofs," Int. J. Electron. Commun. Bio-Eng. (IJECBE), vol. 3, no. 3, 2025.
2. A. S. S. Ahmed et al., "The Future of Electronic Voting System Using Blockchain," Int. J. Adv. Res. Sci., Commun. Technol. (IJARSCT), vol. 3, no. 1, Feb. 2023.
3. Democratic Innovation: Systematic Evaluation of Blockchain-Based Electronic Voting (2022–2025)," Technologies (MDPI), vol. 14, no. 95, 2025.
4. S. K. Singh and A. Gupta, "EtherVote: A Secure Smart Contract-based e-Voting System," ResearchGate, 2026.
5. "Optimizing the EVM with Internet and Blockchain technology: A systematic review for enhancing secured e-Voting," J. Comput. Sci., vol. 4, no. 1, 2025.
6. G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper, 2024.
7. R. Kumar and V. Saini, ”HAC-Bchain: A Secure and Scalable Blockchain-Shard Based E-Voting System,” in Proc. ICCCT, pp. 67–72, 2023.
8. A. Balti et al., ”A Decentralized and Immutable E-Voting System Using Blockchain,” in Proc. ICSCSS, pp. 1434–1440, 2023.
9. S. Ghobadi and M. Tavana, ”Blockchain Technology and E-Voting: Opportunities and Challenges,” J. of Information Technology, vol. 38, no. 1, 2022.
10. J. Li, X. Xie, and Y. Cheng, ”Decentralized Privacy-Preserving Solutions for E-Voting,” J. of Cryptographic Engineering, vol. 9, no. 3, 2023.
11. L. Luu and Z. Wang, ”Hybrid Consensus Models for Scalable Blockchain Applications,” ACM Trans. on Blockchain, vol. 5, no. 2, 2023.
12. I. Ahmad and M. Ahmed, ”Cryptographic Advances for Enhanced Privacy in Blockchain- Based Voting,” J. of Cryptology, vol. 35, no. 2, 2022.
13. R. Brown, Y. Li, and S. Kim, ”LLMs for Smart Contract Analysis and Generation: Opportunities and Challenges,” arXiv preprint, 2023.
14. S. Gao et al., ”An anti-quantum e-voting protocol in blockchain with audit function,” IEEE Access, vol. 7, 2019.
15. M. J. M. Chowdhury et al., “Blockchain-based biometric voting system,” Journal of Network and Computer Applications, vol. 190, 2021.