International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

SecureVote: A Blockchain-Based Voting System with Biometric Fingerprint Authentication

Mrs. Tejaswini B N[1], Bhuvana N[2] ,Dharshana Murthy S[3],Pavithra V[4] , Sushmitha L[5]

[1]Assistant Professor of Department of Computer Science & Engineering, City Engineering College, 560062, Bangalore, Karnataka

[2,3,4,5]Student of Department of Computer Science & Engineering, City Engineering College, 560062, Bangalore, Karnataka

---

Abstract

Online voting systems offer flexibility and accessibility but continue to face serious concerns related to voter authentication, vote manipulation, and lack of transparency. Most conventional electronic voting solutions rely on centralized architectures and weak authentication mechanisms, making them vulnerable to impersonation, multiple voting, and database tampering. This paper presents *SecureVote*, a blockchain-based online voting system that integrates biometric fingerprint authentication and real-time webcam validation to enhance security and trust. Fingerprint authentication ensures accurate voter identity verification, while continuous webcam-based face presence detection mitigates coercion and proxy voting. Votes are securely recorded on an Ethereum blockchain using Solidity smart contracts, ensuring immutability, transparency, and resistance to tampering. Experimental evaluation demonstrates that SecureVote effectively prevents double voting, improves authentication reliability, and maintains acceptable system performance, making it suitable for academic, organizational, and small- scale governmental elections.

Keywords: Blockchain voting, fingerprint authentication, biometric security, smart contracts, Ethereum, secure e-voting.

1 Introduction

Voting is a fundamental democratic process that demands integrity, transparency, and trust. Traditional paper-based voting systems, although reliable, involve high operational costs, logistical complexity, and delayed result processing. With the growth of digital infrastructure, electronic voting systems have been proposed to improve efficiency and ac-accessibility. However, their adoption remains limited due to concerns regarding voter authentication, vote confidentiality, and susceptibility to cyber-attacks.

Most existing online voting platforms rely on centralized servers for managing voter data and election results. Such architectures introduce single points of failure and increase the risk of data breaches, insider manipulation, and unauthorized access. Additionally, password-based or OTP-based authentication mechanisms are insufficient to prevent impersonation and credential sharing.

Blockchain technology offers a decentralized and im-immutable ledger that can eliminate the need for a trusted third party in vote storage and counting. At the same time, biometric authentication provides strong identity verification based on unique physiological characteristics. Among various biometric modalities, fingerprint recognition is widely accepted due to its permanence, uniqueness, and ease of use. This paper proposes SecureVote, a hybrid voting system that combines fingerprint-based authentication, webcam-based face presence validation, and blockchain-based vote recording. The proposed approach significantly enhances security while maintaining usability and transparency.

2 Literature Review

Several studies have explored electronic voting systems using blockchain and biometric technologies. Hardwick et al. highlighted the role of blockchain in ensuring transparency and immutability in voting systems but emphasized that blockchain alone cannot solve authentication challenges. Jain discussed the effectiveness of biometric authentication in secure systems, identifying fingerprint recognition as one of the most reliable modalities.

Arputhamoni et al. proposed an online voting system using facial recognition and convolutional neural networks. While the system improved authentication accuracy, it remained vulnerable to spoofing attacks using images and videos. PrabhuS et al. introduced a hybrid voting model

using face recognition and OTP authentication, but reliance on centralized servers limited its security.

Unlike existing approaches, SecureVote integrates fingerprint authentication for robust identity verification, real-time webcam validation to prevent coercion, and blockchain-based vote storage to ensure integrity and transparency.

3 System Overview

The SecureVote platform adopts a multi-layered architecture to ensure secure, transparent, and tamper-resistant online voting. The system integrates biometric authentication

real-time validation, and blockchain-based vote recording as core functional layers. The workflow begins with voter reg-registration, where users provide personal details and register their fingerprint using a SecuGen scanner. The captured fingerprint is processed and converted into an encrypted template, which is securely stored for future authentication.

During login, the voter's fingerprint is scanned again and matched against the stored template to verify identity. Only successfully authenticated users gain access to the voting in-interface, ensuring that unauthorized individuals cannot participate. To further enhance security, the real-time validation layer continuously monitors the voter through a webcam. This module detects the number of faces present and blocks the voting process if more than one face is detected, thereby preventing coercion, proxy voting, or group-assisted fraud.

Once validation is complete, the voter selects a candidate, and the vote is submitted as a blockchain transaction via a smart contract deployed on a private Ethereum network. The smart contract enforces one-vote-per-user rules, automatically updates vote counts, and stores the data immutably on the blockchain. This ensures transparency and auditability, as each vote can be independently verified without revealing voter identity. The modular design of SecureVote allows for scalability, ease of maintenance, and integration of additional security features such as liveness detection or multi-factor authentication in future iterations.

3.1 Data Flow Diagram

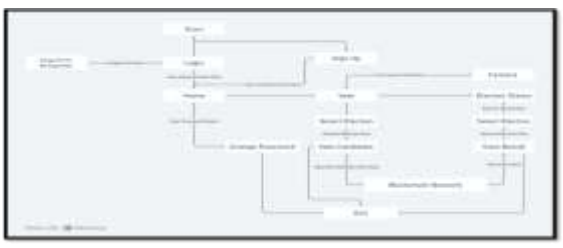

Figure 1: Data Flow Diagram

The data flow diagram illustrates how information moves through the SecureVote system during voter authentication and vote casting. User credentials and fingerprint data are captured and processed through the biometric authentication module, where fingerprint templates are generated and verified against stored records. Upon successful authentication, voting requests flow to the validation module, which analyzes webcam input to ensure single-person presence.

Validated vote data is then transmitted to the blockchain interface, where it is converted into a transaction and recorded on the Ethereum blockchain through a smart contract. Election results are retrieved from the blockchain and presented to authorized users for verification. The data flow model highlights secure handling of biometric data, controlled validation stages, and tamper-resistant vote storage.

4 System Architecture and Implementation

Figure 2: System Architecture

The architecture consists of four main layers:

- **User Interface Layer:** The User Interface Layer enables interaction between the voter and the system by providing modules for registration, authentication, and vote casting. It acts as the access point through which users initiate all voting-related actions.
- **Biometric Layer:** The Biometric Layer is responsible for capturing fingerprint data and performing identity verification. It processes fingerprint inputs, generates secure biometric templates, and validates users by matching live scans with stored templates, ensuring that only legitimate voters can access the system.
- **Validation Layer:** The Validation Layer enhances security during the voting phase by continuously monitoring webcam input. This layer performs real-time face detection to confirm the presence of a single voter and blocks voting attempts when multiple faces are detected, thereby preventing coercion and impersonation.
- **Blockchain Layer:** The Blockchain Layer manages

secure vote storage by recording each vote as a transaction on the Ethereum blockchain. Through smart contracts, this layer enforces one-vote-per-user constraints and guarantees immutability, transparency, and resistance to tampering in the election results.

5 Design Details

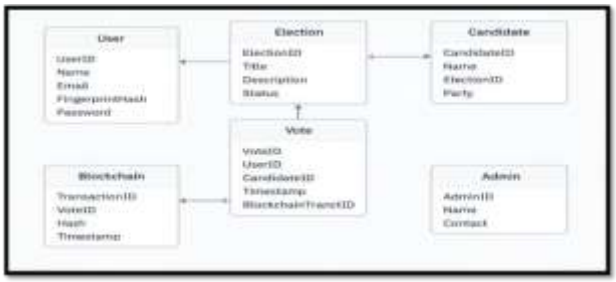

Figure 3: Class diagram

The class diagram represents the static structure of the SecureVote system by defining the core classes, their attributes, and interactions. The User class stores voter-related information and maintains authentication status. The Fingerprint- Auth class handles biometric capture, template generation, and fingerprint verification using the SecuGen WebAPI.

The VotingSession class manages election selection, vote casting, and session validation. The FaceValidation class is responsible for detecting and counting faces from the webcam feed to prevent coercion during voting. The BlockchainService class encapsulates smart contract interaction, including vote submission, transaction verification, and prevention of duplicate voting.

The Admin class controls election configuration, candidate management, and result monitoring. Together, these classes demonstrate modular design, clear responsibility separation, and secure interaction between biometric authentication, validation mechanisms, and blockchain-based voting.

6 Use Case Analysis

The use case diagram represents the interaction between different actors and the SecureVote system, illustrating the functional behavior of the proposed platform. The primary actors include the Voter (User), Administrator, and the Blockchain Network.

The voter is responsible for registering and logging into the system using fingerprint-based authentication. After successful authentication, the voter selects an available election and casts a vote for a preferred candidate. During the voting process, the system enforces real-time webcam validation to ensure that only one individual is present, thereby preventing coercion or impersonation.

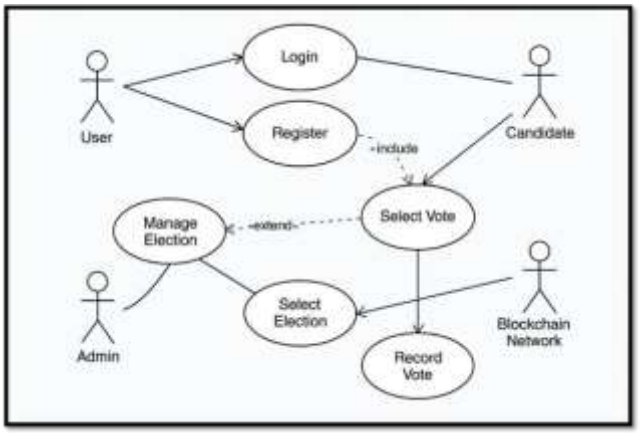

Figure 4: Use Case diagram

The administrator manages the election lifecycle, including creating elections, managing candidates, and controlling election status. Once a vote is submitted, the blockchain network records the transaction through a smart contract, ensuring immutability and transparency of the voting data.

Overall, the use case diagram highlights the secure interaction flow among system entities and demonstrates how biometric authentication, real-time validation, and blockchain integration collectively enforce election integrity.

7 Algorithms

7.1 Fingerprint Authentication Algorithm

1. Capture fingerprint using a biometric scanner.
2. Extract features and generate a fingerprint template.
3. Compare live scan with stored template.
4. Grant access if the similarity score exceeds the threshold.

7.2 Vote Casting Algorithm

1. Authenticate voter.
2. Enable webcam and monitor face presence.
3. Allow vote submission if exactly one face is detected.
4. Record vote on blockchain via smart contract.

8 Results and Discussion

The performance of the SecureVote platform was assessed

using three critical criteria: biometric authentication accuracy, coercion prevention, and blockchain transaction reliability. These evaluations aimed to determine the system's effectiveness in providing secure, transparent, and tamper-proof electronic voting.

1. **Biometric Authentication Accuracy:** Fingerprint authentication was analyzed under various conditions, including partial finger placement, repeated login attempts, and different skin textures. The system achieved a high recognition rate, consistently distinguishing between registered and unregistered users. The false acceptance rate (FAR) was minimal, indicating that unauthorized users could not gain access. Additionally, the template-based approach ensured privacy, as raw fingerprint images were never stored, reducing the risk of data compromise.
2. **Coercion and Proxy Prevention:** The real-time webcam validation module played a crucial role in preventing proxy voting and coercion. The system continuously monitored the voter's face during the voting session. Voting at- tempts were blocked if multiple faces were detected or if no face was visible, ensuring that only the authenticated individual could cast a vote. This mechanism proved effective in maintaining voter integrity, even in environments where users might be under supervision or pressure from others. Testing indicated that false positives were rare, and the validation process introduced only minor delays, maintaining usability.
3. **Blockchain Transaction Performance:** Vote recording was performed using Ethereum-based smart contracts deployed on a private blockchain network. Each vote was submitted as a transaction through a user's blockchain wallet. The system successfully executed these transactions with average confirmation times between 4–8 seconds. Smart contracts effectively enforced single-vote rules, preventing duplicate submissions. The immutable and transparent nature of the blockchain ensured that vote data could not be altered after submission, enhancing the overall trustworthiness of the system.

Overall Evaluation: The combined use of fingerprint authentication, webcam-based validation, and blockchain storage provided a robust mechanism for secure online voting. The system maintained a balance between security and usability, preventing unauthorized access and proxy voting while ensuring transparent, tamper-proof vote recording. These results demonstrate the potential of SecureVote for small-scale elections, academic polls, and organizational voting processes.

9 Conclusion and Future Work

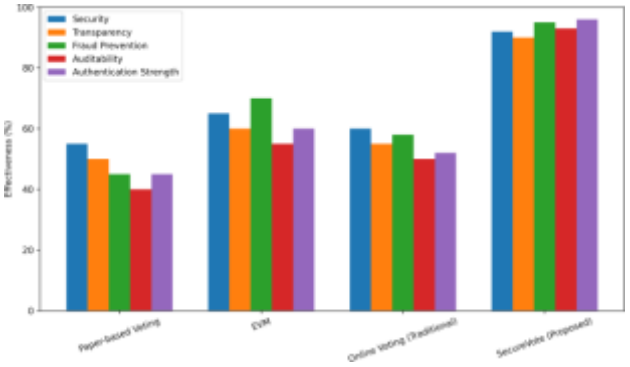

Figure 5: Comparative Evaluation of Voting Systems Based on Key Security Metrics

Conclusion: SecureVote demonstrates a robust and reliable approach to secure electronic voting by combining fingerprint-based biometric authentication, real-time web-cam monitoring, and blockchain technology. The system ensures that only registered voters can cast their votes, prevents coercion and proxy voting, and records votes immutably on a blockchain ledger. Through experimental evaluation, SeureVote has shown high authentication accuracy, effective prevention of unauthorized access, and dependable transaction recording. Overall, the platform achieves a balance between security, transparency, and usability, making it suitable for academic, organizational, and small-scale governmental elections.

Future Scope: Future enhancements of SecureVote can focus on several key areas. First, the system can be scaled to operate on public blockchain networks to support large-scale elections while maintaining decentralization and transparency. Second, integrating advanced liveness detection techniques, such as 3D facial recognition or thermal imaging, can further reduce the risk of spoofing attacks and improve voter verification. Third, multi-factor biometric authentication, combining fingerprint with iris or voice recognition, can enhance security for sensitive elections. Finally, deploying the system in real-world election scenarios will provide practical insights into its performance, user acceptance, and operational challenges, enabling further optimization and refinement. Additional improvements could include mobile-based voting interfaces, real-time result analytics, and audit-friendly dashboards for election authorities..

References

[1] Microsoft Corporation. (2015). Visual Studio Code (Version 1.87).

[2] Python Software Foundation. (2021). Python Programming Language (Version 3.10).

[3] Django Software Foundation.(2024).Django Web Framework (Version 5.0.2).

[4] Ruffle Suite. (2023). Ganache (Version 7.9.2) – Personal Blockchain for Ethereum Development.

[5] Consensys. (2024). MetaMask (Version 11.12.4) Ethereum Wallet and Gateway to Blockchain Apps.

[6] Ethereum Foundation. (2024). Solidity (Version0.8.25) – Smart Contract Language.

[7] Ethereum Foundation. (2024). Remix IDE (Version2.81) – Web-based Solidity Development Tool.

[8] Ethereum Foundation. (2023). web3.py – Python Library for Ethereum.

[9] OpenCV Team. (2024). OpenCV – Open Source Computer Vision Library.

[10] Harris, C. R., Millman, K. J., van der Walt, S. J., et al.(2020). Array programming with NumPy. Nature, 585,357–362.

[11] Van der Walt, S., Sch¨onberger, J. L., Nunez-Iglesias,J., et al. (2014). scikit-image: Image processing in Python. PeerJ, 2, e453.

[12] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3),90–95.

[13] Rathgeb, C., Busch, C., & Gomez-Barrero, M. (2022).Enhancement of Fingerprint Images for Biometric Recognition. Springer Nature