International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 265-272 March 2026-Sept 2026 www.csjournals.com

Blockchain for Decentralized Voting System: Preventing Double Voting with Consensus Mechanism

Neha
Chaudhary *Ranbir Singh University, Haryana, India*

Dr. Sneha
Chaudhary *Ranbir Singh University, Haryana, India*

Dr. Anupam Bhatia
Chaudhary Ranbir Singh University, Haryana, India

Abstract *— Every democratic election is based on a simple promise: every eligible voter gets exactly one vote. In India, this promise has not always been delivered by paper ballots and traditional electronic voting machines, with charges of duplicate voter rolls, proxy voting and tampering emerging repeatedly. We propose a voting system based on fingerprint biometric authentication and blockchain ledger to minimize such risks. A voter's identity is checked one time, during a one-time registration step. On election day, a voter provides a fingerprint scan and doesn't have to show any documents again. A Solidity smart contract implements a hasVoted flag, which prevents any wallet from voting more than once. Personal information lives off-chain in a separate database. The public ledger thus proves that a legitimate vote was made, but not who. There are three points where double voting is caught – a fingerprint match, a backend status check and the smart contract itself – so an attacker who beats one layer still has two more to beat. We built a working prototype using Go for the backend, Solidity smart contracts on a local Hardhat network, and Python for the fingerprint sensor. The paper also reviews five recent blockchain-voting proposals, places our framework alongside them, and lays out what still needs to happen before a system like this could be trusted at national scale.*

Index Terms *— Blockchain, Biometric Authentication, Smart Contract, Electronic Voting, Double-Voting Prevention, Decentralization, Off-chain Storage.*

I. INTRODUCTION

Voting is how citizens choose their representatives, and the whole exercise only works if every vote is counted once, and counted correctly. Paper ballots and standalone electronic voting machines both still lean on a central authority to store and manage voter data. That one point creates a single point of failure – if the central record is changed or duplicated or is just out of date, the fairness of the entire election can be called into question.

India has witnessed this play out more than once. During the 2018 Telangana Assembly elections, thousands of names of the voters were found to be duplicated or missing, and the opposition parties had accused the state government of tampering with the electoral rolls [1]. Similar controversy over voter lists took place in the 2015 Bihar Assembly elections [2] and, in the 2021 West Bengal Assembly elections, major parties accused each other of fake voting and voter intimidation in several constituencies [3]. The Electronic Voting Machines have also faced scrutiny with parties questioning their reliability since 2009 and one party staging a public demonstration of a mock EVM tampering scenario in 2017 [4]. However, the Election Commission of India insists that no large-scale tampering has ever been proved [5].

Blockchain technology provides a solution to many of these problems at once. A blockchain ledger is not stored on one computer, but copied to a number of independent computers. So if you wanted to change a vote after it has been recorded, you would have to change the same record on a majority of those computers simultaneously – exactly what consensus mechanisms are designed to prevent. Add a smart contract that automatically checks whether a voter has already voted. Add biometric authentication that links each vote to a real person enrolled in the system. The result is a system that relies much less on any single trusted party. In this paper we describe such

a framework, the rationale of the layered architecture and the state of the current prototype with respect to similar work in the literature.

II. PRELIMINARIES

A. Blockchain

A blockchain is a distributed ledger that can only be appended to. Nakamoto first realized it at scale as the basis for Bitcoin [6]. Data is grouped into blocks. Each block stores the cryptographic hash of the previous block. Any change to a past entry changes every hash after it, which is obvious to the rest of the network immediately [7]. Each block can include a set of validated voting transactions, a timestamp, and a link to the previous block. The first block, the genesis block, does not have a parent.

B. Biometric Authentication

Fingerprint authentication involves matching a live scan to a template captured during enrolment. A fingerprint is difficult to fake or transfer to another person. So here it is only used to verify that the person voting is the same person that registered, not to know who they voted for.

C. Smart Contracts

A smart contract is a piece of program logic deployed on the blockchain. Ours was written in Solidity for an Ethereum-compatible network. Every node executes the same contract code in the same way. If the rule was "if hasVoted is already true, reject this vote", it could not be silently bypassed for one voter without every other node noticing that something was different.

III. RELATED WORK

Table I summarises five recent proposals that share the same broad goal, moving elections onto a blockchain, but differ sharply in how they handle consensus, privacy, and duplicate-vote prevention.

Ref.	Year	Core Idea	Technical Approach
[8]	2019	A consortium blockchain built for a single day of polling rather than an always-on cryptocurrency network.	Presiding officers create blocks by hashing officer ID with a random seed; blocks are sealed at poll closure using a Merkle-tree pairing.
[9]	2022	A full voting management system that treats a vote almost like spendable currency.	Each voter receives one non-replenishable Voting Coin; a Chain Security Algorithm revalidates the hash chain on every new block.
[10]	2020	A framework-as-a-service so an election commission can plug into infrastructure it already owns.	Hyperledger Fabric handles ledger mechanics; a microservices layer and REST APIs connect it to end-user applications.
[11]	2022	A working prototype ("BlockVote") with separate contracts for voting logic and one-time codes.	Two chaincodes on Hyperledger Fabric; private data collections keep voting codes visible only to the organising body.
[12]	2023	A systematic literature survey spanning more than 40 years of e-voting research.	Builds a standardised set of criteria (accuracy, eligibility, privacy, verifiability, robustness, uniqueness) and scores prior work against it.

Table I. Summary of related blockchain voting proposals

Shahzad and Crowcroft [8] make a useful point: a blockchain built for cryptocurrency doesn't fit an election out of the box. Proof-of-work and similar mechanisms assume a long-running, adversarial network, while a poll runs for a single day with a known set of officials, so their answer is to replace mining with a lighter block-sealing step. Farooq et al. [9] are more software engineering oriented. They treat a vote almost like a spendable coin, so that casting it automatically empties the wallet - a clean, physical way to block a second vote, rather than relying only on a database flag. Mukherjee et al. [10] do not focus on cryptography but on deployment, and propose a layered service that an election commission could plug into existing infrastructure. Poniszewska-Maranda et al. [11] go the furthest with implementation where they split voting logic and one time codes across different smart contracts and use private data collections so that voting codes are not visible to other organisations sharing the network. Almeida et al. [12] do not contribute anything new, but survey the field. Their conclusion: privacy and verifiability receive consistent attention in the literature, but eligibility and real-world testing do not receive nearly as much attention -- a gap that informed several choices in the framework below.

Two weaknesses emerge again and again in this body of work, and both influenced our design. None of the systems we examined have actually been tested in the conditions of a large-scale election, so performance claims are from lab settings rather than real-world deployments. Second, most "decentralised" designs still route voter eligibility through a centralised authority before a vote ever reaches the chain. Decentralisation removes a single point of failure from vote counting, sure, but the same problem just moves one step earlier — to the question of who's allowed to vote in the first place.

IV. PROBLEM STATEMENT

Existing voting systems, paper-based or electronic, remain open to voter impersonation, duplicate voting, and unauthorised changes to stored records. Manually checking documents during voter verification is slow and error-prone at the scale of a national election, and keeping voter data in one central database creates a single target whose compromise affects everyone at once. The problem this paper takes on is how to verify a voter's identity, make sure that voter can cast only one vote, and keep the resulting record tamper-resistant, all without repeatedly exposing sensitive personal information across the network.

V. CONTRIBUTION

This work contributes a voting framework that combines biometric authentication with blockchain-enforced voting rules. The main pieces:

Biometric-gated voting. A fingerprint match is required before any voting transaction is generated, tying every vote to a genuinely enrolled person without re-collecting identity documents on election day.

Layered double-voting defense. Duplicate votes are caught at three independent points, biometric match, backend status check, and the smart contract's hasVoted flag, so no single point of failure can let a second vote through on its own.

On-chain / off-chain separation. Personally identifying information never touches the blockchain; only the wallet address, vote, and timestamp go on-chain, which keeps the immutable ledger free of data that privacy law may eventually require to be corrected or deleted.

Decentralised validation. Votes are validated by the network consensus mechanism as opposed to a single centralised server so no one node can unilaterally alter a recorded vote.

International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 265-272 March 2026-Sept 2026 www.csjournals.com

Transparent verifiability. Vote transactions are visible to every node, so election results can be independently recomputed by any authorised party without revealing which candidate any individual voter chose.

VI. PROPOSED FRAMEWORK

The framework is organised into five layers: voter-facing, biometric authentication, backend processing, smart contract, and blockchain network, shown in Fig. 1. Splitting the system this way lets each layer be reasoned about and secured on its own, and it means no single layer alone decides whether a vote is accepted.

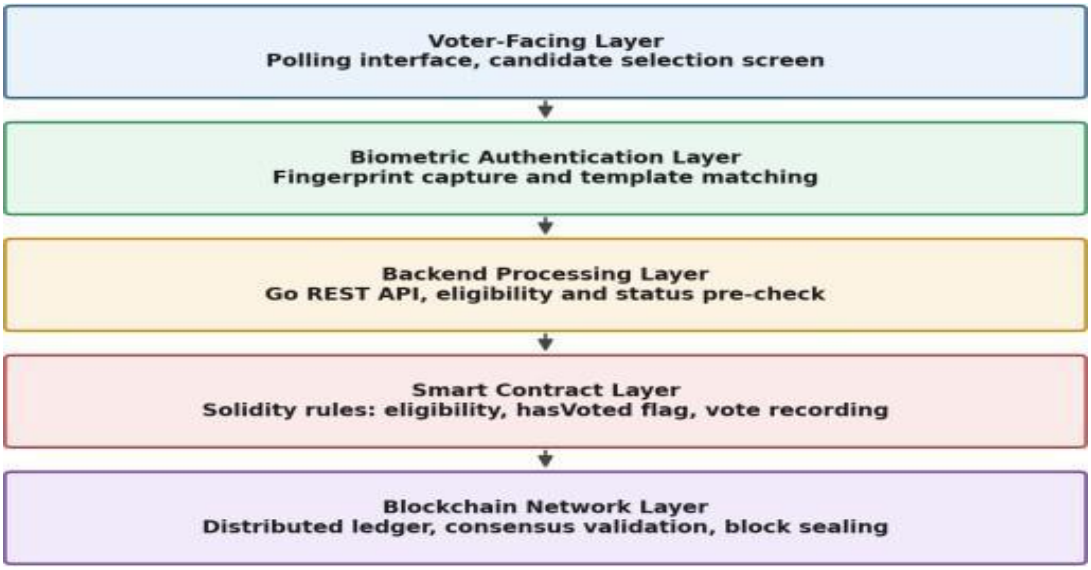

Fig. 1. Five-layer architecture of the proposed voting framework

A. Notation

Table II lists the notation used in the registration and voting algorithms that follow.

Notation	Description
A	Registration administrator
V	Voter
FID	Fingerprint template identifier generated
W	Blockchain wallet address assigned to a voter
CID	Candidate identifier
HV	hasVoted flag stored in the smart contract (t / f)
Tx	Vote transaction submitted to the smart contract
H()	SHA-256 hash function

Table II. Algorithm notation

B. Voter Registration (One-Time)

Registration happens once per voter, before any election takes place, and is the only point in the system where identity documents get collected.

1. Voter V submits identity documents (Voter ID / Aadhaar, mobile number, email) to administrator A.

2. A checks the documents and if the check fails, the application is rejected.

3. Upon approval, a fingerprint is taken and enrolled to generate V's FID.

4. A wallet address W is generated and linked to V only in the off-chain database — this mapping never gets written to the blockchain.

C. Voting Process (Per Election)

On election day, the voter is asked for nothing but a fingerprint scan. No documents get requested a second time; doing so would defeat the point of having a separate registration phase.

1. V scans a fingerprint; the sensor matches it against the enrolled template and retrieves the existing FID.

2. If no match is found, the vote is rejected and a retry is offered.

3. The backend does a quick first-pass check of V's current voting status and returns immediate feedback.

4. The transaction Tx including W and the selected CID is sent to the smart contract.

5. The contract verifies that W is registered, and that HV = false. If HV is true, the transaction is rolled back.

6. If the check passes, the vote is recorded against CID and HV is updated to true, blocking any further attempt from W.

D. Layered Defense Against Double-Voting

We use three independent checks instead of one to block double voting. The biometric layer prevents someone from voting under someone else's identity, the backend layer is a fast, first-line status check, and the smart contract layer provides the final, network-verified assurance that can't be changed post-facto.

These are listed in order from easiest to hardest to attack, with spoofing a fingerprint being difficult but not impossible, compromising of a backend server being harder, and rewriting consensus across a distributed network being, by design, the hardest of the three. Anyone who makes it past the first layer still has to make it past two more difficult ones before they can record a duplicate vote.

International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 265-272 March 2026-Sept 2026 www.csjournals.com

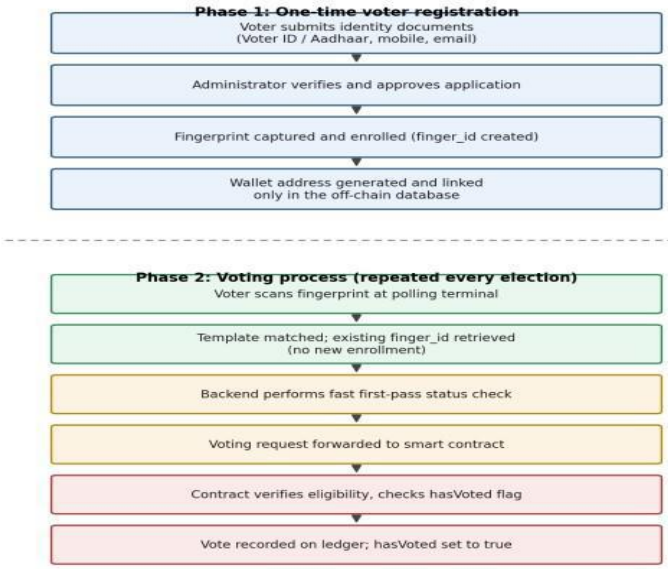

Fig. 2. Registration and voting workflow

E. On-Chain and Off-Chain Data Separation

Personal and identifying data like name, Voter ID/Aadhaar number, mobile number, email and the mapping of a person to his/her wallet remains completely off-chain. Only the minimum required for vote integrity is stored on the blockchain, the wallet address, the hasVoted status, the vote itself and a timestamp, as shown in Fig. 3. And this split does two things at the same time: personal data can still be corrected or deleted to satisfy privacy obligations, and a full public read of the ledger shows that votes were cast and counted correctly without ever revealing who cast which one.

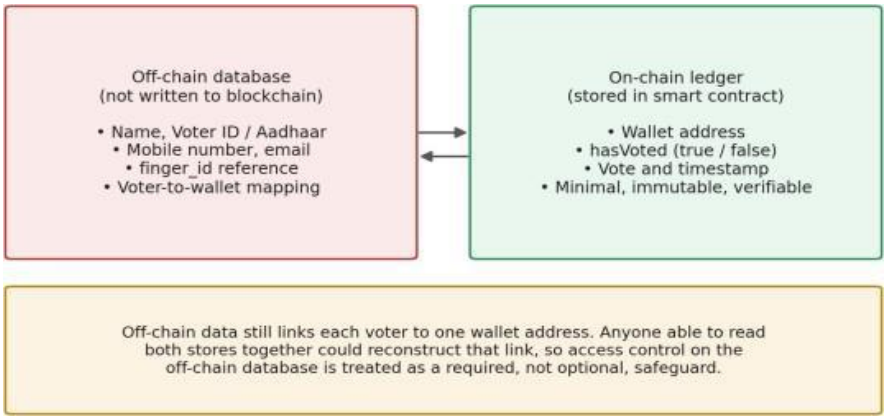

Fig. 3. Decoupling of on-chain and off-chain voter data

F. Implementation Stack

The prototype implements the backend REST API using Go mainly due to the way it tackles concurrency under many simultaneous authentication requests, Solidity for the smart contract as it is the most widely audited language for Ethereum-compatible contracts, Python for fingerprint sensor integration as it has a mature SDK

support, and Hardhat for local blockchain testing so that contract logic can be checked before any deployment decision is made.

VII. SECURITY ANALYSIS AND EVALUATION CRITERIA

Instead of testing the framework merely on whether a demo runs, we evaluate it based on five criteria, using the classification employed in recent survey work [12]:

Accuracy: does the system count each valid vote and reject each duplicate attempt?

Verification of eligibility: Are only registered and verified voters allowed to vote?

Privacy: can we link a vote to the voter who cast it without having access to both on-chain and off-chain data at the same time?

Verifiability: Can a voter or an auditor verify that a vote was correctly recorded without being able to change it?

Resilience: What happens to a vote in progress if the fingerprint scan fails, the network drops, or the backend goes down during the transaction?

The first four are built into the current prototype: the smart contract's hasVoted check ensures accuracy, wallet registration ensures eligibility, the off-chain/on-chain split ensures privacy, and the shared ledger ensures verifiability. The most brittle part at the moment is resilience under failure – Fig. 2 describes the success path in detail but does not yet describe what should happen if the network drops between the backend check and the call to the smart contract. We leave that for future work.

Also worth being honest about is a limitation this framework shares with almost everything reviewed in Section III: eliminating a single point of failure from vote counting does not eliminate it from vote eligibility. The off-chain database still maps each voter to one wallet address. The block-level data itself does not contain information about who cast which vote, but someone with access to both the off-chain database and the on-chain ledger could, in principle, reconstruct that link. So controlling access to the off-chain database isn't optional here; it's a required security control.

VIII. CONCLUSION AND FUTURE SCOPE

This paper presented a blockchain-based voting framework that combines fingerprint biometric authentication, smart contracts, and a decentralised ledger to cut down on voter impersonation, duplicate voting, and unauthorised changes to election records. Double voting is caught by three independent checks rather than one, and personal information stays off-chain, so the public ledger only ever shows that a valid, unique vote was recorded. We implemented a prototype using Go, Solidity, Python, and Hardhat, and positioned it against five recent blockchain-voting proposals from the literature.

A few extensions are planned. Decentralised document storage through IPFS, paired with off-chain attribute-based encryption for fine-grained access control over sensitive voter documents [13], could let election authorities verify eligibility documents without any single central database holding them in the clear. Zero-knowledge proofs could push voter anonymity further than the current on-chain/off-chain split already does. And the framework still needs to be tested at something closer to real constituency scale, under realistic network conditions, before any claim about its readiness for a live election can be made with real confidence.

REFERENCES

[1] The Hindu, "Telangana voter list controversy: Allegations of deletion and duplication of voters," The Hindu, Oct. 2018.

[2] India Today, "Bihar elections and voter list irregularities raise concerns," India Today, November 2015.

[3] BBC News, "West Bengal polls marred by violence and fake voting allegations," BBC News, Apr. 2021.

[4] The Indian Express, "AAP demonstrates alleged EVM tampering in Delhi Assembly," The Indian Express, Aug. 2017.

[5] Election Commission of India, "Status paper on Electronic Voting Machines and VVPAT," New Delhi, India, 2017.

[6] Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[7] C. Subramanian, A. A. George, et al., "The Blockchain Technology."

[8] B. Shahzad and J. Crowcroft, "Trustworthy electronic voting using adjusted blockchain technology," IEEE Access, vol. 7, pp. 24477-24488, 2019.

[9] M. S. Farooq, U. Iftikhar, and A. Khelifi, "A framework for making voting system transparent using blockchain technology," IEEE Access, vol. 10, 2022.

[10] P. P. Mukherjee, A. A. Boshra, et al., "A hyper-ledger fabric framework as a service for improved quality e-voting system," in Proc. IEEE Region 10 Symp. (TENSYMP), Dhaka, Bangladesh, Jun. 2020.

[11] A. Poniszewska-Maranda et al., "Decentralized electronic voting system based on Hyperledger Fabric," in Proc. IEEE International Conf. Services Computing (SCC), 2022.

[12] R. L. Almeida, F. Baiardi, D. Di Francesco Maesa, L. Ricci, "Impact of decentralization on electronic voting systems: A systematic literature survey," IEEE Access, vol. 11, 2023.

[13] S. Chaudhary and A. Bhatia, "A blockchain-based framework for securing digital documents using off-chain attribute-based encryption," in Proc. 2nd Int. Conf. Computational and Characterization Techniques in Engineering & Sciences (IC3TES), Lucknow, India, Nov. 2024.

[14] S. Wang, X. Qu, Q. Hu, X. Wang, and X. Cheng, "An uncertainty and collusion proof voting consensus mechanism in blockchain," IEEE/ACM Trans. Netw., vol. 31, no. 5, pp. 1751–1764, Oct. 2023.