ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 11, Issue 3, March 2026

Secure E-Voting System with End-to-End Verifiability

[1] Sonika, [2] Srividhya, [3] Monika,K, [4] Egneswari
[1] [2] [3] [4] Dept. of Computer Science and Engineering Er. Perumal Manimekalai College of Engineering (Autonomous), Hosur, Tamil Nadu, India

Abstract: ***The growing adoption of digital platforms for institutional decision-making demands election systems that are simultaneously secure, transparent, and privacy-preserving. This paper presents a Secure E-Voting System with End-to-End Verifiability, a software-only platform designed for small-scale institutional elections such as college student councils, professional associations, and community governance bodies. The system enforces complete voter anonymity through strict database-level identity separation. Vote confidentiality is achieved using the Elliptic Curve Integrated Encryption Scheme (ECIES), combining ECC NIST P-256 key encapsulation with AES-256-GCM authenticated encryption. Every cast vote is immutably appended as a block to a custom Python blockchain governed by Proof of Work (PoW) consensus, producing a tamper-evident, publicly auditable ledger. Voter access is gated by two-phase authentication combining Bcrypt password hashing and a time-limited OTP. The proposed system is fully aligned with UN Sustainable Development Goal (SDG) 16: Peace, Justice, and Strong Institutions. Per-vote encryption latency is under 200 ms, suitable for real-time interactive use.***

Index Terms: E-Voting, Blockchain, ECIES, ECC NIST P-256, AES-256-GCM, End-to-End Verifiability, Voter Anonymity, Proof of Work, Two-Factor Authentication, SDG 16, Identity Separation, Cryptographic Protocols.

I. INTRODUCTION

Voting is the foundational mechanism of democratic governance, employed at every level from national elections to local club committees. Traditional paper-based systems are susceptible to ballot stuffing, miscounting, inadequate audit trails, and geographic inaccessibility. The transition to electronic voting (e-voting) promises to resolve these weaknesses, yet the vast majority of deployed e-voting systems merely digitise the counting process without providing cryptographic guarantees of correctness. Voters are asked to trust an opaque system rather than being given the means to verify it [1].

A credible e-voting system must simultaneously satisfy four non-negotiable properties. First, voter anonymity: it must be computationally infeasible to associate any stored vote record with the voter who submitted it. Second, integrity: votes must be recorded immutably, and any attempt at post-hoc tampering must be immediately and publicly detectable. Third, end-to-end verifiability: individual voters and independent observers must be able to confirm, through publicly available evidence, that the election outcome correctly reflects the cast votes. Fourth, accessibility: the system must operate through a standard web browser on commodity hardware, without special software or devices.

This paper presents a system that satisfies all four properties through the principled combination of four cryptographic and architectural mechanisms: (i) ECIES hybrid encryption for vote confidentiality; (ii) a custom Proof of Work blockchain for vote integrity and auditability; (iii) strict database-level identity separation for voter anonymity; and (iv) two-phase Bcrypt-OTP authentication for access control. The system is implemented in software using Python (Flask), SQLite, and PyCryptodome, with a responsive HTML/CSS/JavaScript frontend.

The system directly advances UN Sustainable Development Goal 16: Peace, Justice, and Strong Institutions, by providing institutions with a low-cost, technically rigorous mechanism for conducting elections that are provably fair, transparent, and accountable.

II. RELATED WORK

The theoretical foundations of secure e-voting date to Chaum's 1981 work introducing untraceable electronic mail and blind signatures as primitives for simultaneously achieving anonymity and verifiability [1]. Fujioka et al. [2] operationalised this duality in a practical blind-signature voting protocol.

Blockchain-based approaches have emerged as a compelling architectural solution for vote integrity. Ayed [3] demonstrated through a conceptual blockchain voting system that distributed ledger immutability can eliminate single points of failure.

McCorry et al. [8] extended this with a smart-contract-based self-tallying protocol on Ethereum. The foundational Proof of Work mechanism was formalised by Nakamoto [4] in the context of Bitcoin.

The Helios web-based open-audit voting system [9] demonstrates that cryptographic verifiability can be delivered through a standard web browser using El-Gamal encryption with a Sako-Kilian mixnet and Chaum-Pedersen decryption proofs. The present work is inspired by Helios's commitment to web accessibility and open auditability, but differs in cryptographic construction—using ECIES on NIST P-256 [5] rather than El-Gamal—and in its architectural approach to anonymity through identity separation rather than mixnets.

Hybrid encryption combining ECC key encapsulation with AES authenticated encryption achieves optimal efficiency and IND-CCA2 security simultaneously [6]. The HKDF key derivation function used in our ECIES implementation is standardised in RFC 5869 [7]. The NIST P-256 elliptic curve is mandated under FIPS 186-4 [5] and is the basis of widely deployed TLS implementations.

III. SYSTEM ARCHITECTURE

The system comprises three integrated functional modules: the Voter Portal, the Admin Dashboard, and the Blockchain and Audit Subsystem. Together they implement the full election lifecycle while enforcing cryptographic security at every stage.

A. Voter Portal

The Voter Portal is the primary interface through which eligible voters participate in elections. It implements a linear, tamper-resistant workflow: (1) registration with a Bcrypt-hashed, salted password; (2) credential verification (Phase 1 of two-phase authentication); (3) OTP delivery and verification (Phase 2), required before any voting action is permitted; (4) display of all active elections with candidate listings and metadata; and (5) submission of an end-to-end encrypted vote, encrypted in-browser using the election's ECC public key before transmission. Upon vote submission, the voter receives a visual participation confirmation. No data record exists anywhere in the system that associates the voter's identity with their submitted vote.

B. Admin Dashboard

The Admin Dashboard provides the election administrator with full lifecycle management and auditability tools: (1) creating, configuring, opening, and closing elections; (2) adding, editing, and removing candidates; (3) monitoring real-time voter participation statistics; (4) decrypting and tallying all votes using the election's ECC private key at closure; and (5) reviewing the Blockchain Ledger and Audit Log. The administrator authentication flow is entirely separate from the voter authentication path, preventing cross-role access.

C. Blockchain and Audit Subsystem

Every submitted vote is appended as a new, cryptographically linked block to a custom Python blockchain. Each block stores the SHA-256 hash of the preceding block, the encrypted vote payload, a Unix timestamp, and a nonce satisfying the Proof of Work condition, making the ledger append-only and tamper-evident. A Blockchain Explorer interface provides real-time inspection of raw block data and chain integrity status. An independent Audit Log records all system events with actor identifiers and timestamps.

IV. METHODOLOGY

A. Vote Encryption Using ECIES

Every vote is encrypted before it leaves the voter's browser session using ECIES. An ephemeral ECC key pair (epriv, Epub) is generated on the NIST P-256 curve. An ECDH shared secret is computed as S = ECDH(epriv, PKelection). A 256-bit AES session key is derived as K = HKDF-SHA256(S, salt, info). The vote plaintext V is encrypted as (C, tag) = AES-256-GCM(K, nonce, V), where the GCM authentication tag provides ciphertext integrity. The bundle {Epub, nonce, C, tag} is transmitted to the server and stored in the votes table. Decryption requires SKelection, held exclusively by the administrator. The GCM tag is verified before any plaintext is accepted, ensuring corrupted ciphertexts are detected and rejected. This construction provides IND-CCA2 security under the ECDDH assumption.

B. Blockchain Immutability via Proof of Work

Each vote is stored as block Bi containing: a Unix timestamp, the encrypted vote payload, the SHA-256 hash of Bi-1, and a nonce n such that SHA-256(Bi-1.hash||payload||n) begins with four zero nibbles (difficulty level 4). Any post-hoc modification to Bi changes Hi, which no longer matches the stored previous-hash in Bi+1. Tamper detection requires only O(N) SHA-256 evaluations.

C. Voter Anonymity via Identity Separation

The voter_participation table stores exactly one attribute per voter: a boolean has_voted flag sufficient to enforce the no-double-voting constraint. The votes table stores exclusively the encrypted ciphertext bundle with no foreign key, no implicit join path, and no timestamp or session correlation. There exists no data structure in the system that maps voter identities to vote records. This anonymity guarantee is information-theoretic for the votes table: even with complete database access and unlimited computation, no algorithm can associate a stored ciphertext with the voter who submitted it.

D. Two-Phase Authentication

In Phase 1, the voter submits a username and password verified against a Bcrypt hash with a unique per-user salt. Bcrypt's configurable cost factor ensures the work factor of offline attacks can be kept ahead of available hardware. In Phase 2, a time-limited 8-character alphanumeric OTP is generated server-side, stored as a SHA-256 hash, and delivered via an out-of-band channel. The OTP subsystem enforces a maximum of five verification attempts per session; exceeding this threshold triggers account lockout. A successful Phase 2 completion is required before the vote submission endpoint becomes accessible. Consequently, a compromised password alone is insufficient to permit fraudulent vote submission.

Table 1. Cryptographic Algorithms Employed in the System

Algorithm	Standard	Role in System
ECC Key Gen.	NIST P-256 (FIPS 186-4)	Election key pair; public key for ECIES vote encryption
ECDH Agreement	NIST SP 800-56A Rev. 3	Ephemeral shared secret derivation
HKDF	RFC 5869 (HMAC-SHA-256)	Derives AES-256 session key from ECDH secret
AES-256-GCM	FIPS 197 / SP 800-38D	Authenticated symmetric encryption
Proof of Work	SHA-256, difficulty-4 nonce	Block mining; tampering computationally infeasible
SHA-256	FIPS 180-4	Block hash chaining; OTP storage
Bcrypt	Blowfish adaptive KDF	Password hashing with per-user salt

V. SYSTEM IMPLEMENTATION

The system is implemented as a software-only solution with no dependency on specialised hardware or proprietary infrastructure, keeping deployment cost low for small institutions. All components run on a standard server with Python 3.

Table 2. System Technology Stack

Component	Technology	Purpose
Backend	Python 3 + Flask	REST API, business logic, Jinja2 rendering

Database	SQLite	Voter records, election metadata, vote storage
Cryptography	PyCryptodome	ECC, AES-256-GCM, HKDF, SHA-256
Password Hash	Bcrypt + Flask-Session	Adaptive password hashing, session tokens
Frontend	HTML5, CSS3, JS	Voter portal and admin dashboard UIs
Prod. Server	Waitress WSGI	Production HTTP request serving
Blockchain	Custom Python	PoW mining, SHA-256 linking, chain verify
Auth OTP	SHA-256 hashed OTP	Time-limited 8-char OTP, rate-limited

A. Voter Portal Interface

The voter-facing home page presents the system's three core guarantees: ECC Encryption, Voter Anonymity, and Transparent Tally, communicating cryptographic assurances to non-expert voters. Following successful two-phase authentication, the voter is directed to the Voter Dashboard, which presents all active elections with configured date ranges and a direct link to the ballot.

B. Administrative Interface

The Admin Dashboard consolidates all election management functions: real-time aggregate statistics, an elections table with live status indicators, and quick-access navigation to the Blockchain Ledger, Audit Log, voter management, and election creation screens. A live participation doughnut chart visualises the voted-versus-pending proportion. The Admin Portal login screen implements an isolated authentication pathway, entirely separate from voter authentication. Credentials, session tokens, and access-control checks are managed independently, preventing any cross-role access.

VI. EVALUATION AND RESULTS

A. Security and Functional Testing

Eight security test cases were executed to verify correct enforcement of all security-critical system behaviours, targeting specific attack vectors. All eight test cases passed, confirming correct enforcement of security properties under adversarial conditions. The identity linkage test confirmed that no query, join, or inference across the database schema can associate a vote record with a voter.

Table 3. Security and Functional Test Results

Test Case	Expected Behaviour	Result
Double-vote attempt	Second submission rejected; event flagged in Audit Log	PASS
Blockchain block tampering	SHA-256 hash mismatch identified by integrity check	PASS
OTP brute-force (>=6 tries)	Account lockout triggered; further attempts blocked	PASS
Decryption with incorrect key	AES-GCM tag verification fails; exception raised	PASS
Identity linkage attempt	No data path from any vote record to any voter identity	PASS
Audit log completeness	All events recorded with timestamp, actor, action type	PASS

Concurrent submissions	DB transaction isolation; no race condition observed	PASS
Admin access without OTP	Vote/admin endpoints inaccessible until Phase 2 passes	PASS

B. Performance Evaluation

Performance measurements were conducted on a standard development machine (Intel Core i5, 8 GB RAM, Python 3.10). ECIES vote encryption—comprising ECDH key agreement, HKDF derivation, and AES-256-GCM encryption—completed in an average of 165 ms per vote. ECIES decryption for post-election tallying averaged 182 ms per vote. Blockchain block mining at difficulty level 4 averaged 280 ms per block. The Flask application handled 50 concurrent simulated voter requests without error, race condition, or data integrity violation.

These figures confirm that the ECIES scheme is suitable for real-time interactive voting at the target deployment scale. The PoW mining latency of 280 ms is operationally acceptable given that votes are submitted sequentially at human interaction speeds. Multithreading would substantially improve throughput for larger-scale deployments.

C. Comparison with Related Systems

Relative to Helios [9], our design differs in three key respects. First, we use ECIES on NIST P-256 rather than El-Gamal with a mixnet, eliminating computationally expensive shuffle and shuffle-proof steps at the cost of not providing universally verifiable tally proofs in the current version. Second, our anonymity mechanism is architectural (identity separation) rather than cryptographic (mixnet permutation). Third, our blockchain ledger provides a persistent, inspectable immutability record absent in Helios.

Table 4. Comparison with Helios Web-Based Voting System [9]

Property	Proposed System	Helios [9]
Vote Encryption	ECIES (ECC P-256 + AES-256-GCM)	El-Gamal (1024-bit)
Anonymity	DB-level identity separation	Sako-Kilian mixnet
Immutability	PoW blockchain	Bulletin board (server)
Verifiability	Audit log + chain integrity	Universal verifiability proof
Tally Proof	Admin-decrypted tally	Chaum-Pedersen ZKP
Authentication	Bcrypt + OTP (2FA)	Email + password
Deployment	Flask/SQLite (software-only)	Python/PostgreSQL
Target Scale	Small institutional	Small institutional

VII. ADVANTAGES AND LIMITATIONS

A. Advantages

- Provable voter anonymity: Identity separation at schema level provides an information-theoretic anonymity guarantee irrespective of the adversary's computational power.
- Tamper-evident immutability: The PoW blockchain makes post-hoc vote modification computationally infeasible and immediately detectable via a single chain-integrity pass.
- Strong access control: Two-phase authentication (Bcrypt + OTP) with rate limiting ensures neither password compromise nor OTP interception alone is sufficient for unauthorised voting.

- Transparent auditability: The Blockchain Explorer and Audit Log provide full institutional transparency without exposing individual vote content.
- No hardware dependency: The system runs on commodity server infrastructure, accessible to institutions without dedicated IT resources.
- SDG 16 alignment: Directly enables the fair, accountable, and transparent institutional processes promoted by SDG 16.

B. Limitations

- Internet connectivity: Stable network connection required; intermittent connectivity can disrupt the voting session.
- Private key custody: The entire vote corpus is protected by a single ECC private key held by the administrator. Key loss is irrecoverable; key compromise breaks vote confidentiality for all votes in that election.
- No universally verifiable tally: No cryptographic proof that the tally was computed correctly over stored ciphertexts; observers must trust the administrator's decryption.
- PoW scalability: At very high submission volumes, PoW mining latency may introduce bottlenecks; alternative consensus mechanisms should be evaluated for larger deployments.

VIII. FUTURE WORK

Several research and engineering directions are identified to address current limitations. The most significant cryptographic enhancement is integration of Paillier partially homomorphic encryption to enable server-side encrypted vote tallying without any individual decryption step, eliminating the private key custody risk entirely.

Zero-Knowledge Proofs (ZKPs)—specifically non-interactive Sigma-protocols using the Fiat-Shamir heuristic—will be incorporated to provide universally verifiable tally proofs, allowing any observer to verify that the published tally is consistent with the stored ciphertexts.

Migration to a permissioned Ethereum or Hyperledger Fabric blockchain would provide decentralised, publicly auditable ledger immutability. TOTP-based two-factor authentication per RFC 6238 will replace the current custom OTP implementation. A mobile application for iOS and Android will be developed to improve voter accessibility. Finally, the system's scalability will be evaluated through load testing and multithreaded block mining will be implemented as required.

IX. CONCLUSION

This paper presented the design, implementation, and systematic evaluation of a Secure E-Voting System with End-to-End Verifiability. The system addresses the four core requirements of credible e-voting—voter anonymity, vote integrity, transparent auditability, and accessibility—through the principled combination of ECIES hybrid encryption, a Proof of Work blockchain, strict database-level identity separation, and two-phase authentication.

All eight security and functional test cases passed, confirming correct enforcement of double-vote prevention, blockchain tamper detection, brute-force lockout, authenticated decryption failure, and complete audit logging. Performance benchmarks demonstrated per-vote encryption latency of 165 ms and block mining latency of 280 ms, confirming suitability for real-time deployment at the intended institutional scale.

The system constitutes a direct and practical contribution to SDG 16 by equipping institutions with a low-cost, cryptographically rigorous mechanism for conducting elections that are provably fair, transparent, and accountable. Future extensions incorporating homomorphic tallying, zero-knowledge tally proofs, and decentralised blockchain infrastructure will advance the system toward the stronger verifiability guarantees of full open-audit voting.

References

[1] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," Commun. ACM, vol. 24, no. 2, pp. 84-90, Feb. 1981.

[2] A. Fujioka, T. Okamoto, and K. Ohta, "A Practical Secret Voting Scheme for Large Scale Elections," in Proc. AUSCRYPT '92, LNCS vol. 718, pp. 244-251, Springer, 1993.

[3] A. B. Ayed, "A Conceptual Secure Blockchain-Based Electronic Voting System," Int. J. Netw. Secur. Appl. (IJNSA), vol. 9, no. 3, pp. 1-9, May 2017.

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

[5] NIST, "Digital Signature Standard (DSS)," FIPS Publication 186-4, National Institute of Standards and Technology, Jul. 2013.

[6] V. S. Miller, "Use of Elliptic Curves in Cryptography," in Proc. CRYPTO '85, LNCS vol. 218, pp. 417-426, Springer, 1986.

[7] H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)," IETF RFC 5869, May 2010.

[8] P. McCorry, S. F. Shahandashti, and F. Hao, "A Smart Contract for Boardroom Voting with Maximum Voter Privacy," in Proc. Financial Cryptography (FC 2017), LNCS vol. 10322, pp. 357-375, Springer, 2017.

[9] B. Adida, "Helios: Web-based Open-Audit Voting," in Proc. 17th USENIX Security Symposium, pp. 335-348, USENIX Association, 2008.

[10] M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption - How to Encrypt with RSA," in Proc. EUROCRYPT '94, LNCS vol. 950, pp. 92-111, Springer, 1995.

[11] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. Pearson Education, 2017.

[12] Legrandin et al., "PyCryptodome: A Self-Contained Python Package of Low-Level Cryptographic Primitives," 2024. [Online]. Available: https://pycryptodome.readthedocs.io

.