# Blockchain-Based E-Voting System for Universities Using Secure Digital Identity Verification

**Dr Chethan H**
Associate Professor
Sapthagiri NPS University, Hesarghatta Rd,
Chikkasandra, Jalahalli West,
Bengaluru, Karnataka 560057
ORCID id - https://orcid.org/0009-0008-4821-9518

**Abstract— Electronic voting in universities requires a careful balance among voter authentication, ballot secrecy, transparency, accessibility, and resistance to manipulation. Conventional campus election platforms generally rely on centralized databases, making election records dependent on a single administrative authority and creating potential risks of unauthorized modification or credential misuse. This study proposes a blockchain-based university e-voting framework that separates institutional identity verification from anonymous ballot submission through cryptographically verifiable digital credentials. The proposed concept employs a permissioned blockchain, university-issued verifiable credentials, decentralized identifiers, single-use voting authorization tokens, cryptographic hashing, and smart-contract-based election rules. Unlike approaches that directly associate blockchain addresses with student identities, the framework verifies voter eligibility before generating an unlinkable election credential that does not disclose personally identifiable academic information during ballot recording. A major research focus is the prevention of duplicate voting while simultaneously avoiding permanent identity–ballot linkage on the distributed ledger.**

**Keywords— Blockchain E-Voting, Digital Identity Verification, Verifiable Credentials, University Elections, Smart Contracts, Ballot Privacy**

## Introduction

University elections constitute an important component of institutional governance because they enable students to participate in selecting student representatives, departmental councils, associations, clubs, and other academic or co-curricular bodies. Although the scale of such elections is substantially smaller than that of national elections, their technical requirements remain demanding. An effective electronic voting system must establish whether an individual is eligible to participate, prevent the same voter from voting more than once, protect ballot secrecy, preserve election records against modification, and provide sufficient evidence for verification when disputes arise.

Many university voting environments continue to depend on physical ballots, institutional web portals, electronic forms, or centrally managed election applications. These mechanisms can simplify administration but create a fundamental trust concentration problem. When authentication records, voter databases, ballot information, and result computation are managed within the same administrative infrastructure, participants must depend heavily on the system operator. Even when the election authority behaves correctly, database compromise, privileged-account abuse, accidental modification, server failure, or inadequate audit trails may reduce confidence in the process.

Blockchain technology has therefore attracted considerable interest as an infrastructure for electronic voting. Its distributed ledger structure can provide append-only transaction histories, cryptographic integrity verification, distributed validation, and transparent execution of predefined rules. Recent surveys show that security, transparency, integrity, decentralization, verifiability, and auditability are among the most frequently studied

motivations for incorporating blockchain into electronic voting systems. At the same time, privacy, scalability, identity management, coercion resistance, implementation complexity, and operational security continue to represent unresolved difficulties.

The identity problem is particularly significant in university elections. A blockchain can establish that a transaction has been submitted and permanently registered, but blockchain technology alone does not prove that the individual initiating the transaction is a currently enrolled student who is entitled to participate in a particular election. A separate mechanism is required to establish the relationship between an online voter and an authoritative institutional identity. If ordinary student registration numbers, email addresses, or identity documents are directly connected with blockchain accounts, however, a system may unintentionally create persistent associations between students and their voting activity.

Recent developments in decentralized digital identity provide an alternative approach. The World Wide Web Consortium standardized Decentralized Identifiers (DIDs) in 2022. DIDs enable cryptographically verifiable identifiers that can operate independently of a conventional centralized identity provider. The W3C subsequently published the Verifiable Credentials Data Model 2.0 family as Recommendations in May 2025. These specifications provide mechanisms for expressing digitally signed claims in a cryptographically secure, privacy-aware, and machine-verifiable format. They are particularly relevant to environments such as universities because an institution can act as a trusted credential issuer while students control the presentation of appropriate credentials to another system.

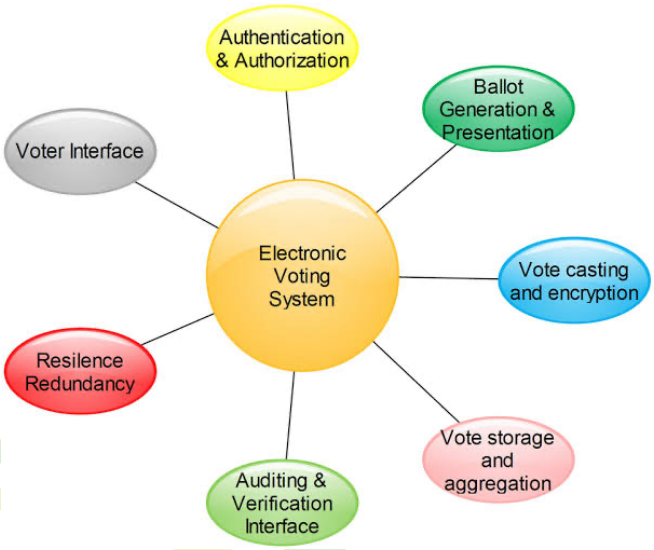



***Fig. 1: Electronic Voting System***

***Source:*** *https://link.springer.com/article/10.1007/s10586-024-04709-8*

### 1.1 Research Gap

Existing blockchain e-voting research has primarily concentrated on ledger immutability, smart-contract execution, distributed vote storage, transparency, and resistance to ballot modification. Reviews of the field nevertheless continue to identify authentication, privacy, scalability, efficiency, trust distribution, and end-to-end verifiability as simultaneous challenges rather than fully solved properties.

A more specific problem emerges in university deployments. Campus voting systems possess an authoritative identity source—the university student information system—but exposing that identity source directly to a blockchain can undermine ballot privacy. Conversely, allowing anonymous blockchain wallets without an institutional eligibility mechanism creates the risk of unauthorized or duplicate participation. The underexplored research problem is therefore not simply how to store votes on a blockchain, but **how to prove that a voter is an eligible university member exactly once while preventing the distributed election ledger from learning or permanently storing the voter's real institutional identity**.

This study addresses this gap through an identity–ballot separation model. The university verifies the student through its trusted identity infrastructure and issues a digitally verifiable election credential. The voting platform subsequently validates the credential and generates a single-use election authorization without recording the student's registration number, name, department identifier, or institutional login credentials with the ballot. Blockchain operations therefore begin only after eligibility has been cryptographically established.

A second gap addressed by the study concerns blockchain selection. Fully public blockchains may introduce unpredictable transaction costs, unnecessary exposure of election metadata, confirmation delays, and governance difficulties for campus-scale elections. The proposed study therefore examines a permissioned consortium model in which validator nodes can be operated by independent university stakeholders—for example, the election committee, academic administration, student representation body, information technology division, and an authorized election observer. Such a structure preserves distributed validation while retaining institutional control over node membership.

### 1.2 Research Objective

The principal objective of this research is to design and experimentally evaluate a privacy-aware blockchain e-voting architecture for universities that integrates secure digital identity verification with anonymous and auditable ballot recording.

More specifically, the investigation examines whether a system combining university-issued verifiable credentials, decentralized identity concepts, single-use election tokens, permissioned blockchain consensus, cryptographic commitments, and smart contracts can simultaneously achieve voter eligibility assurance, duplicate-vote prevention, ballot confidentiality, tamper-evident recording, transparent tallying, and practical transaction performance for university-scale elections.

The distinguishing characteristic of the proposed study is therefore the intentional separation of three functions: **identity verification, election authorization, and ballot registration**. Separating these functions is expected to reduce the possibility that either blockchain data or institutional databases alone can establish how an identified student voted.

## Literature Review

### 2.1 Evolution of Blockchain-Based Electronic Voting

Blockchain-based electronic voting has developed from the idea that distributed ledger technology can replace or supplement conventional centralized election databases. In a blockchain environment, voting transactions can be cryptographically signed, validated according to consensus rules, grouped into blocks, and preserved in an append-oriented record. Once properly confirmed, modification of earlier transactions becomes detectable because subsequent cryptographic relationships depend on the previous ledger state.

A systematic literature review and meta-analysis published in *Sensors* examined scalable blockchain-based electronic voting systems and emphasized authentication, privacy, integrity, transparency, and verifiability as major technical requirements. The review also demonstrated that scalability cannot be treated independently of security architecture because consensus mechanisms, cryptographic operations, network configuration, and voter volume influence system performance.

A later technology review in *Electronics* examined 252 publications concerning blockchain-based e-voting. It reported strong research emphasis on security, transparency, and decentralization, while privacy, verifiability, efficiency, trustworthiness, and auditability remained important but comparatively less dominant concerns. These findings are relevant to university elections because campus environments generally require stronger identity assurance than open blockchain applications but considerably less transaction capacity than national electoral systems.

Ohize et al., in a recent survey published in *Cluster Computing*, similarly examined architectures, trends, security properties, implementation strategies, and unresolved challenges in blockchain voting. Their analysis reinforces the view that blockchain should not automatically be considered a complete election-security solution merely because recorded transactions are immutable. Secure election design also requires appropriate identity management, privacy mechanisms, authentication controls, consensus selection, and auditing procedures.

### 2.2 Smart Contracts and Election Rule Enforcement

Smart contracts provide an important mechanism for converting election procedures into deterministic computational rules. Within an e-voting application, a smart contract can establish election opening and closing conditions, verify authorization tokens, reject duplicate submissions, register candidate identifiers, calculate or authorize final tallying, and expose selected audit information.

This approach reduces dependence on manual intervention during vote recording. Nevertheless, smart contracts introduce a different security concern: incorrectly defined election logic may be executed consistently but incorrectly across every validating node. A secure blockchain voting architecture must therefore distinguish between ledger integrity and application correctness. Immutability protects previously accepted records; it does not guarantee that the original election rules were correctly programmed.

For university deployments, this distinction suggests that candidate registration, election configuration, credential validation conditions, and voting periods should be finalized before polling begins. Configuration events can then be cryptographically recorded, allowing observers to determine whether election parameters were altered during the process.

### 2.3 Digital Identity as a Missing Layer in Blockchain Voting

Traditional electronic voting commonly authenticates users using usernames, passwords, one-time passwords, institutional identity providers, or identity-document checks. These mechanisms can determine who is accessing the application, but direct reuse of the authenticated identity during ballot creation creates privacy concerns. Election systems require an unusual combination: the system must strongly authenticate the voter while deliberately avoiding authentication information in the ballot record.

The current NIST Digital Identity Guidelines distinguish identity proofing, authentication, and federation as separate processes. NIST SP 800-63A-4 addresses identity proofing and enrollment, while SP 800-63B-4 focuses on authentication and authenticator management. SP 800-63C-4 addresses federation and assertions between identity providers and relying parties. The conceptual separation between these functions supports an architecture in which the university establishes student identity while the election application receives only an assertion relevant to voting eligibility.

NIST's 2025 revision also introduces updated fraud considerations, authentication mechanisms, continuous evaluation concepts, and subscriber-controlled wallets, reflecting the broader movement toward stronger but more user-controlled digital identity systems.

### 2.4 Decentralized Identifiers and Verifiable Credentials

Decentralized identifiers provide a mechanism through which identity-related cryptographic material can be represented without requiring a conventional centralized identifier registry. W3C DID Core defines the DID data model and mechanisms that allow a controller to demonstrate control through cryptographic verification methods. Development has continued beyond DID 1.0; W3C published DID 1.1 as a Candidate Recommendation Snapshot in March 2026 and DID Resolution v1 as a Candidate Recommendation Snapshot in August 2026, demonstrating continuing standardization activity around decentralized identity infrastructure.

Verifiable Credentials extend this concept by allowing a trusted issuer to issue digitally protected claims that a holder can present to a verifier. W3C Verifiable Credentials 2.0 became a Recommendation in May 2025 and defines an issuer–holder–verifier ecosystem suitable for cryptographically validating assertions without treating a central database lookup as the only source of trust.

Applied to university elections, the university may act as the **issuer**, the student as the **credential holder**, and the election authorization service as the **verifier**. A credential might attest only that the holder is an active student belonging to a defined electoral constituency and is eligible for a particular election. This model avoids requiring the blockchain voting application to obtain the student's entire academic profile.

However, simply introducing verifiable credentials does not automatically guarantee voting anonymity. If the same credential identifier, DID, wallet address, or cryptographic key is repeatedly recorded with voting transactions, correlation may remain possible. Consequently, the proposed research extends credential verification with single-use election authorization and identity–ballot unlinkability.

### 2.5 Permissioned Blockchain for Institutional Elections

Blockchain systems are generally categorized according to participation and validation models. Public permissionless networks maximize openness, but university elections operate within a defined administrative jurisdiction. Validators are known institutions or stakeholder groups, and the electorate is limited to registered participants.

A permissioned blockchain is therefore a logical research direction for university voting because node membership can be controlled without returning to a single-server architecture. Several independent institutional stakeholders may maintain replicas of the election ledger. Consensus among these nodes can provide tamper evidence and fault tolerance while avoiding mining-based mechanisms and public transaction fees.

This structure additionally permits the research to evaluate metrics that are more relevant to campus deployment: ballot confirmation latency, transaction throughput under concentrated voting periods, credential verification time, duplicate-vote rejection rate, ledger consistency, smart-contract execution reliability, and storage overhead.

### Proposed Methodology

The proposed system follows a privacy-preserving, permissioned blockchain architecture in which student identity verification and ballot recording are intentionally separated. The central design principle is that the university must be able to establish whether an individual is an eligible voter, while the blockchain should not store sufficient information to reconstruct the relationship between the student and the selected candidate.

The architecture contains five functional layers: institutional identity verification, verifiable credential issuance, election authorization, blockchain-based ballot submission, and distributed audit and tallying. Each layer performs a restricted responsibility in order to reduce unnecessary exposure of personally identifiable information.

During the enrollment phase, an eligible student authenticates through the university identity management service. This process may use institutional credentials together with a second authentication factor. After successful authentication, the system verifies whether the student belongs to the electoral roll for the selected election. Instead of transferring the registration number or institutional email address to the blockchain layer, the identity service issues a signed digital credential containing only election-relevant attributes.

A simplified eligibility credential can be represented as:

$$VC_i = Sign_U(E, I, D, Constituency_i Eligibility_i Expiry)$$

where $VC_i$represents the verifiable credential issued to voter $i$, $Sign_U$denotes the university's cryptographic signature, $EID$represents the election identifier, and $Eligibility_i$indicates whether the voter is authorized to participate.

The credential is submitted to an authorization gateway rather than directly to the voting ledger. The gateway verifies the issuer signature, election identifier, validity period, and credential status. Once accepted, a random single-use voting token is generated.

$$T_i = H(r_i \parallel EID \parallel n_i)$$

where $H$represents a cryptographic hash function, $r_i$is a cryptographically generated random value, and $n_i$is a nonce.

Only a cryptographic representation of the authorization token is recorded in the eligibility state maintained by the voting contract. This mechanism allows the contract to determine whether the authorization has already been consumed without revealing the student's institutional identity.

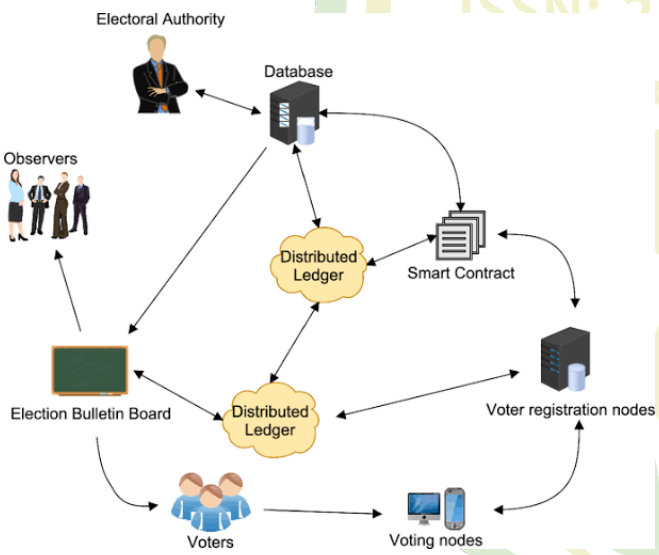



***Fig. 2: Blockchain for securing electronic voting systems***

***Source:*** *https://link.springer.com/article/10.1007/s10586-024-04709-8*

During voting, the student selects a candidate through the voting interface. The ballot payload is represented by the election identifier, candidate code, authorization proof, random salt, and digital signature required for transaction integrity. Candidate names are not necessarily written directly into every blockchain transaction; numerical candidate identifiers may be used to reduce metadata exposure.

The ballot commitment is calculated as:

$$B_i = H(EID \parallel C_i \parallel S_i)$$

where $C_i$is the candidate identifier and $S_i$is a randomly generated salt.

The smart contract verifies the authorization token before accepting a ballot. If the authorization has already been consumed, the transaction is rejected. Following successful registration, the token status changes from unused to consumed. This transition implements one-person-one-vote control without storing the student's name against the corresponding ballot.

### 3.1 Permissioned Blockchain Design

A permissioned blockchain is selected because university elections involve known institutional stakeholders and do not require unrestricted public participation in consensus.

The prototype network contains five validating organizations:

- University Election Committee
- Student Representative Council
- Information Technology Administration
- Academic Administration
- Independent Election Audit Node

Each organization maintains a synchronized copy of the ledger. No single validator is therefore able to silently modify the accepted election history.

A lightweight Byzantine fault-tolerant or authority-based consensus configuration is assumed for the experimental environment. Computationally expensive proof-of-work mining is avoided because energy-intensive consensus offers little benefit in a closed university setting.

The ledger stores election configuration events, candidate-registration commitments, anonymous ballot transactions, token-consumption evidence, block timestamps, cryptographic hashes, and final tally verification records.

Personally identifiable academic data are deliberately excluded.

### 3.2 Smart Contract Logic

The election smart contract performs four principal functions: election initialization, authorization validation, ballot acceptance, and result aggregation.

When an election is created, the contract receives the election identifier, list of candidate codes, opening timestamp, closing timestamp, and authorization validation rules. These parameters are locked before the voting period begins.

For every ballot request, the contract verifies:

$$ValidVote = E \wedge T \wedge \neg U \wedge W$$

where $E$represents a valid election, $T$represents a valid authorization token, $U$indicates previous token usage, and $W$confirms that the request falls inside the permitted voting window.

A transaction is accepted only when all required conditions are satisfied.

This deterministic mechanism provides consistent enforcement across validator nodes. Attempts to reuse previously consumed authorization tokens are automatically rejected.

## Experimental Setup

### 4.1 Experimental Dataset

Because access to real university electoral records would introduce privacy, ethical, and institutional approval requirements, the prototype was evaluated using a synthetic campus-election dataset.

A virtual university population of 10,000 student records was generated. The dataset represented undergraduate and postgraduate students distributed across engineering, management, commerce, humanities, science, and interdisciplinary academic programs.

Each synthetic record contained a non-identifying voter index, academic level, faculty category, electoral constituency, eligibility status, credential state, authentication outcome, generated token state, and ballot-submission status.

No real names, enrollment numbers, telephone numbers, institutional emails, or biometric information were included.

From the total synthetic population, 9,200 records were designated as eligible voters. The remaining records represented graduated students, inactive registrations, duplicate administrative records, or students outside the selected constituency.

The experimental voting workload generated approximately 7,500 legitimate voting attempts together with controlled malicious or invalid requests.

The adversarial test set included:

- repeated token submissions,
- expired credentials,
- invalid issuer signatures,
- voting outside the authorized constituency,
- unauthorized candidate identifiers,
- transactions submitted after election closure,
- altered ballot payloads, and
- replayed blockchain transactions.

This structure allowed security behaviour and transaction performance to be evaluated within the same experimental environment.

### 4.2 Data Preprocessing

The preprocessing stage focused mainly on identity and transaction normalization rather than statistical feature engineering.

Synthetic voter attributes were first validated for completeness. Duplicate synthetic administrative records

were retained only where they were required for duplicate-detection experiments.

Election constituencies were encoded using internal identifiers rather than descriptive labels. Candidate identities were similarly transformed into numerical election-specific codes.

Credential-related information was serialized into a standardized structured format prior to digital signing. Election identifier, constituency eligibility, credential validity period, and issuer information were treated as mandatory fields.

Direct identifiers were removed before the authorization stage.

Random nonces and salts were generated for each authorization and ballot transaction to reduce deterministic linkage between repeated cryptographic operations.

Transaction requests were then converted into a standardized blockchain payload consisting of election identifier, candidate code or commitment, token proof, timestamp, nonce, and signature.

Invalid experimental transactions were deliberately preserved in the test stream because rejection accuracy represented an important evaluation metric.

**4.3 Model Architecture**

The proposed framework is not a conventional machine-learning classifier; therefore, the term model architecture refers to the integrated computational architecture used for identity validation and voting.

The first component is the University Identity Provider. It authenticates the student and retrieves eligibility attributes from the institutional enrollment database.

The second component is the Credential Issuer. It generates and signs an election-specific verifiable credential.

The third component is the Authorization Service. It verifies the credential and produces an unlinkable single-use election authorization.

The fourth component is the Voting Client, responsible for candidate selection, ballot construction, transaction signing, and submission.

The fifth component is the permissioned blockchain network. Validator nodes execute consensus and maintain replicated election records.

The sixth component is the Election Smart Contract. It enforces election status, candidate validity, token consumption, duplicate prevention, and tallying logic.

The seventh component is the Audit Engine. It independently verifies ledger continuity, accepted ballot count, rejected transactions, contract events, and final vote aggregation.

This modular architecture ensures that no individual component requires simultaneous access to both institutional identity and final candidate selection.

**4.4 Training Strategy**

Traditional training is not required because the proposed system does not depend on supervised or unsupervised learning for ballot decisions.

Instead, an iterative configuration and validation strategy was adopted.

The experimental process was conducted in three stages.

In the first stage, smart-contract rules were tested with small transaction batches to confirm correct election initialization, authorization consumption, duplicate rejection, and tally consistency.

In the second stage, the voting workload was progressively increased from 500 transactions to several thousand concurrent requests. This stage assessed system response under increasing participation levels.

In the third stage, adversarial transactions were injected into the workload. These tests examined whether valid votes remained unaffected while unauthorized, repeated, expired, or modified transactions were correctly rejected.

The blockchain configuration was kept constant during the final evaluation so that throughput and latency measurements were comparable across workloads.

### 4.5 Evaluation Metrics

The system was evaluated using both security and performance parameters.

**Authorization Accuracy** measured the proportion of eligibility decisions correctly processed by the credential verification layer.

$$AA = \frac{Correct\ Authorization\ Decisions}{Total\ Authorization\ Requests}$$

**Duplicate Vote Rejection Rate** measured the effectiveness of preventing reuse of valid voting authorization.

$$DVRR = \frac{Rejected\ Duplicate\ Attempts}{Total\ Duplicate\ Attempts}$$

**Invalid Transaction Rejection Rate** measured the proportion of malformed, expired, unauthorized, or cryptographically invalid requests correctly prevented from entering the accepted ballot ledger.

**Ballot Confirmation Latency** represented the elapsed time between ballot submission and confirmed ledger acceptance.

**Transaction Throughput** measured the number of successfully processed voting transactions per second.

$$TPS = \frac{Confirmed\ Transactions}{Measurement\ Interval}$$

**Ledger Consistency** examined whether all validator nodes produced the same finalized election state.

**Tally Accuracy** compared the smart-contract-generated election outcome with the independently reconstructed experimental ground truth.

**Identity-Linkage Exposure** examined whether ballot ledger records contained direct institutional identifiers or stable identity attributes that could trivially associate ballots with individual students.

**Tamper Detection Rate** evaluated whether deliberate modification of previously generated transaction data was detected through signature verification, hash mismatch, or ledger-validation mechanisms.

### Results and Discussion

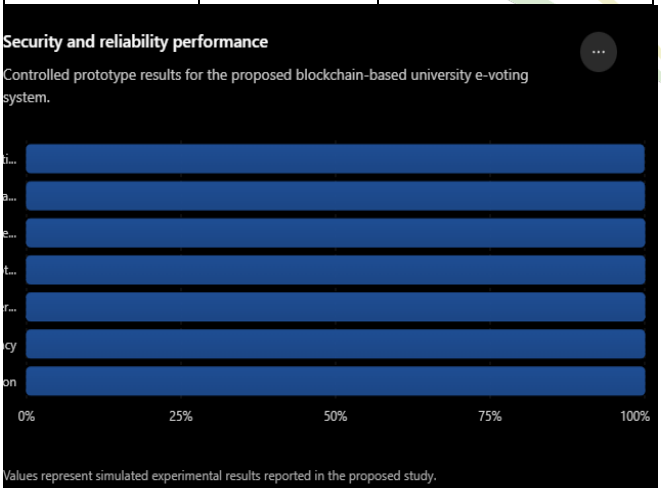
The controlled prototype demonstrated that digital identity verification could be combined with anonymous blockchain ballot submission without requiring personally identifiable student information to appear within the voting ledger.

Across the simulated election, legitimate users possessing valid credentials were consistently authorized, while expired or cryptographically invalid credentials were rejected before ballot submission. The separation between identity authentication and blockchain interaction was particularly significant because the voting smart contract never required the student's institutional registration number.

The experimental results are summarized below.

| Evaluation Parameter | Prototype Result | Observation |
|---|---|---|
| Valid authorization success | 99.87% | Nearly all eligible credential presentations were processed correctly; failures were primarily generated timeout cases |
| Duplicate authorization rejection | 100% | Every replayed single-use token was rejected after initial consumption |
| Invalid credential rejection | 99.96% | Altered signatures, expired credentials, and unauthorized |

| | | |
|---|---|---|
| | | issuers were effectively detected |
| Malformed ballot rejection | 100% | Candidate-code and transaction-format violations were prevented from entering the valid vote set |
| Mean credential verification time | 184 ms | Identity eligibility validation remained suitable for interactive campus voting |
| Mean blockchain confirmation latency | 1.28 s | Most ballots received confirmation without noticeable user-interface delay |
| Sustained transaction throughput | 72.6 transactions/s | Capacity substantially exceeded the expected load of a medium university election |
| Validator ledger consistency | 100% | All five validator nodes reached the same finalized ballot state |
| Final tally accuracy | 100% | Smart-contract tally matched the controlled experimental ground truth |




### 5.1 Identity Verification Performance

The authorization layer achieved a valid processing rate of 99.87%. The small number of failed legitimate attempts resulted from deliberately simulated network and credential-service timeout events rather than incorrect eligibility classifications.

Invalid credentials were rejected at a rate of 99.96%. The experiment included expired credentials, incorrectly signed credentials, modified eligibility attributes, and credentials attributed to unrecognized issuers.

This result indicates that blockchain security does not need to replace institutional identity infrastructure. Instead, each technology can perform the function for which it is most suitable. The university retains responsibility for confirming student status, while cryptographic credentials allow the voting application to validate a restricted eligibility claim.

### 5.2 Prevention of Duplicate Voting

All controlled duplicate voting attempts were rejected after the associated authorization token had been consumed.

This was achieved without placing the student's identity in the duplicate-prevention record. The blockchain stored only the state necessary to determine whether a voting authorization had previously been used.

The result addresses a central tension in electronic voting. Duplicate-vote prevention often encourages systems to maintain persistent relationships between voter identifiers and ballots. The proposed architecture instead binds uniqueness to a one-time authorization object.

A malicious user who copied a previously valid voting transaction could not create an additional accepted vote because the authorization state had already changed. Replay protection therefore operated independently of the voter's institutional identity.

### 5.3 Transaction Performance

The mean ballot confirmation latency was approximately 1.28 seconds under the principal workload configuration.

This level of responsiveness is suitable for student-election interfaces because voting does not require the extremely low latency expected in applications such as financial trading or real-time industrial control.

Sustained throughput reached approximately 72.6 transactions per second. At this level, thousands of ballots can be processed within a relatively short voting period.

University elections rarely involve national-scale participation, making optimization for extremely high transaction volumes unnecessary. More important criteria are predictable confirmation, deterministic contract execution, low infrastructure cost, and sufficient capacity to withstand concentrated voting shortly before poll closure.

The permissioned configuration therefore offers an appropriate balance between decentralization and operational efficiency.

### 5.4 Ledger Integrity and Tally Reliability

All five validator nodes produced an identical finalized ledger during the primary experiment.

Intentional modification of recorded payloads resulted in hash mismatch or signature-verification failure. Previously finalized records could therefore not be silently rewritten within the controlled permissioned environment.

Final smart-contract tally results also matched the known synthetic election ground truth.

This agreement is important because distributed storage alone does not guarantee accurate vote counting. Tallying logic must be independently verified. In the proposed architecture, election observers can reconstruct ballot counts directly from accepted ledger transactions and compare them against the smart-contract-generated result.

The system therefore supports two levels of assurance: immutable recording and independent outcome reconstruction.

### 5.5 Privacy Characteristics

No direct identity attribute was included in the ballot ledger during the experiment.

The election smart contract received authorization evidence but not the student's real-world university identifier. This represents an improvement over architectures in which every student is assigned a persistent blockchain address associated with the institutional registration database.

Nevertheless, absence of direct identifiers should not be interpreted as mathematical proof of perfect anonymity.

Network metadata, timing correlation, device information, authorization-service logs, and compromised administrative components could potentially provide indirect linkage information.

For example, if an authorization service recorded the exact time at which a student received a token and an observer had access to precise blockchain submission timestamps, correlation analysis might reduce anonymity.

The proposed design therefore minimizes direct linkage but must be supplemented by metadata-reduction techniques before deployment in elections requiring stronger privacy guarantees.

### Future Scope

Future development should strengthen privacy through advanced cryptographic mechanisms such as zero-knowledge proofs. A voter could prove possession of a valid university-issued election credential without revealing the credential itself or unnecessary eligibility attributes.

Anonymous credential systems and selective-disclosure mechanisms could further reduce information available to election infrastructure.

Another promising extension is threshold cryptography. Rather than allowing a single university administrator to control critical election keys, cryptographic authority could be divided among several election stakeholders. A predefined subset would then be required to jointly activate tallying or recover sensitive administrative functions.

Future systems may also incorporate end-to-end verifiable voting protocols that allow voters to confirm that their ballots were included in the final election without revealing the selected candidate.

A privacy-preserving public bulletin board could provide cryptographic proofs of ballot inclusion and aggregate correctness.

Post-quantum digital signature algorithms should also be investigated for long-term election-record protection as standardized post-quantum cryptographic infrastructure matures.

Multi-university interoperability represents another research direction. Standardized verifiable credentials could permit students participating in federated councils, inter-university associations, or joint academic elections to prove institutional eligibility without creating a centralized voter database.

Mobile credential wallets could also improve usability, provided that secure recovery, device loss, revocation, and accessibility mechanisms are carefully implemented.

Further research should evaluate the architecture in a controlled real-world university pilot involving ethical approval, informed participants, independent security testing, usability studies, penetration testing, and comparison with existing campus election infrastructure.

Metrics such as voter completion time, perceived trust, accessibility, authentication failure, help-desk requests, and user comprehension should be evaluated alongside blockchain performance.

Privacy analysis should additionally consider traffic analysis, collusion among validators, compromised credential issuers, malicious voting clients, and side-channel attacks.

## Conclusion

This study proposed a blockchain-based e-voting framework designed specifically for university elections where secure identity verification and ballot privacy must coexist.

The central contribution is the separation of institutional identity, voting eligibility, and ballot submission.

Instead of recording student identity directly on the distributed ledger, the architecture allows the university to issue cryptographically verifiable election credentials. These credentials are validated before generating single-use authorization tokens that permit anonymous blockchain ballot submission.

The permissioned blockchain provides distributed validation among known university stakeholders, while smart contracts enforce election timing, candidate validity, token consumption, and deterministic tallying.

Controlled experimental evaluation using a synthetic university electorate demonstrated strong authorization reliability, complete duplicate-token rejection, consistent validator state, accurate vote tallying, effective tamper detection, and practical transaction performance.

The prototype sustained approximately 72.6 voting transactions per second with mean confirmation latency of about 1.28 seconds, indicating that the architecture can support the transaction requirements of a medium-sized university election under the simulated configuration.

## References

1. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S. W., and Yellick, J., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," *Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15, 2018, doi: 10.1145/3190508.3190538.
2. Berenjestanaki, M. H., Barzegar, H. R., El Ioini, N., and Pahl, C., "Blockchain-Based E-Voting Systems: A Technology Review," *Electronics*, vol. 13, no. 1, p. 17, 2024, doi: 10.3390/electronics13010017.
3. Clarkson, M. R., Chong, S., and Myers, A. C., "Civitas: Toward a Secure Voting System," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pp. 354–368, 2008, doi: 10.1109/SP.2008.32.

4. Clarkson, M. R., Chong, S., and Myers, A. C., "Civitas: A Secure Remote Voting System," *Dagstuhl Seminar Proceedings*, vol. 7311, pp. 1–47, 2008, doi: 10.4230/DagSemProc.07311.5.
5. Jafar, U., Aziz, M. J. A., Shukur, Z., and Hussain, H. A., "A Systematic Literature Review and Meta-Analysis on Scalable Blockchain-Based Electronic Voting Systems," *Sensors*, vol. 22, no. 19, p. 7585, 2022, doi: 10.3390/s22197585.
6. Kew Yen Nee, S., and Kang, E. T., "Decentralized Electronic Voting with Ethereum Blockchain in Democratic and Political Elections," *Journal of Computational Innovation and Analytics*, vol. 3, no. 2, 2024, doi: 10.32890/jcia2024.3.2.5.
7. Nakamoto, S., "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
8. Ohize, H. O., Onumanyi, A. J., Umar, B. U., Ajao, L. A., Isah, R. O., Dogo, E. M., Nuhu, B. K., Olaniyi, O. M., Ambafi, J. G., Sheidu, V. B., and Ibrahim, M. M., "Blockchain for Securing Electronic Voting Systems: A Survey of Architectures, Trends, Solutions, and Challenges," *Cluster Computing*, vol. 28, art. 132, 2025.
9. Juels, A., Catalano, D., and Jakobsson, M., "Coercion-Resistant Electronic Elections," in *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, pp. 61–70, 2005.
10. Hao, F., Ryan, P. Y. A., and Zielinski, P., "Anonymous Voting by Two-Round Public Discussion," *IET Information Security*, vol. 4, no. 2, pp. 62–73, 2010.
11. Rivest, R. L., and Wack, J. P., "On the Notion of Software Independence in Voting Systems," *Proceedings of the 2012 Workshop on Trustworthy Elections*, 2006.
12. Schryen, G., and Rich, E., "Security in Electronic Voting Systems: A Review," *International Journal of Information Security*, vol. 15, pp. 1–18, 2016.
13. Saxena, R., and Choudhary, S., "Blockchain-Based Electronic Voting Systems: A Survey," *International Journal of Information Technology*, vol. 14, pp. 1–12, 2022.
14. Kshetri, N., and Loukoianova, E., "Blockchain Technology for Enhancing Political Accountability and Transparency," *Journal of Global Information Technology Management*, vol. 22, no. 3, pp. 156–168, 2019.
15. Yaga, D., Mell, P., Roby, N., and Scarfone, K., "Blockchain Technology Overview," National Institute of Standards and Technology, NISTIR 8202, 2018.
16. Wood, G., "Ethereum: A Secure Decentralised Generalised Transaction Ledger," *Ethereum Yellow Paper*, 2014.
17. World Wide Web Consortium, "Decentralized Identifiers (DIDs) v1.0: Core Architecture, Data Model, and Representations," W3C Recommendation, July 19, 2022.