International Journal of Innovative Research in Engineering
Volume 7, Issue 2 (March - April 2026), PP: 132-144.
www.theijire.com ***ISSN No: 2582-8746***

Decentralized Blockchain-Based Framework for Secure Digital Voting Applications

S A Althaf Ahammed[1], V Harinisree[2], F Ayisha Begum[3], P S Afreena[4], R K Nalan[5]
[1] *Assistant Professor, Department of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, Tamilnadu, India.*
[2,3,4,5]*Department of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, Tamilnadu, India.*

How to cite this paper:
S A Althaf Ahammed[1], V Harinisree[2], F Ayisha Begum[3], P S Afreena[4], R K Nalan[5], "Decentralized Blockchain-Based Framework for Secure Digital Voting Applications", IJIRE-V7I2-132-144.

Abstract: *Electronic voting systems have gained significant attention as a modern alternative to traditional paper-based electoral processes. However, existing electronic voting mechanisms often rely on centralized infrastructures that introduce vulnerabilities such as data tampering, insider manipulation, and lack of audit transparency. These limitations reduce public trust in digital election systems. This paper presents a secure blockchain-based electronic voting platform with cryptographic verification designed to ensure transparency, integrity, voter anonymity, and tamper detection. The proposed system implements a custom lightweight blockchain architecture where all electoral events are recorded as cryptographically linked blocks using SHA-256 hashing. Smart rule enforcement logic ensures voter eligibility verification, prevents duplicate voting, and maintains role based access control. SQLite is integrated for participation tracking while preserving vote anonymity by separating voter identity from ballot payload. The system architecture consists of user interface, logic control layer, blockchain ledger, and persistent storage. Experimental evaluation demonstrates realtime tamper detection, efficient vote recording, secure block validation, and scalability for localized deployments. The results confirm that the proposed model enhances electoral integrity, reduces operational cost, and strengthens trust in digital democratic processes.*

Keywords: *Blockchain, Electronic Voting, Cryptographic Verification, SHA-256, Tamper Detection, Decentralized Ledger, Secure E-Voting.*

I.INTRODUCTION

Electronic voting (e-Voting) systems employ electronic devices and computational infrastructures to cast and count votes in electoral processes [1], [2]. Compared to traditional paper-based voting, electronic systems offer advantages such as faster result computation, reduced administrative cost, and improved accessibility for large-scale elections [3]. These benefits have motivated governments and institutions worldwide to explore digital voting infrastructures [4].

Despite these improvements, most existing e-Voting implementations rely on centralized architectures where vote records are stored within a single database or controlled server environment [5]. Such centralized models introduce single-point-of-failure vulnerabilities and increase the risk of insider manipulation or cyber-attacks [6]. Security analyses of deployed electronic voting systems have revealed concerns regarding limited transparency and restricted public auditability [7].

To address these limitations, researchers have investigated cryptographic approaches and distributed ledger technologies for secure electoral systems [8], [9]. Blockchain technology, in particular, provides a tamper-evident and append-only ledger structure in which records are cryptographically linked using secure hash functions [10]. This mechanism ensures that once a vote is recorded, any attempt to alter historical data becomes computationally detectable [11].

Recent studies indicate that permissioned and lightweight blockchain frameworks are more suitable for institutional elections than public cryptocurrency networks due to reduced computational overhead and improved deployment flexibility [12], [13]. Additionally, integrating structured validation logic with blockchain logging enhances duplicate vote prevention and voter eligibility enforcement [14].

In this paper, a secure blockchain-based electronic voting platform with cryptographic verification is proposed. The system integrates a custom blockchain ledger, SHA-256based hashing, role-based authentication control, and structured participation tracking. Unlike public blockchain voting solutions, the proposed architecture focuses on institutional deployment, operational efficiency, and practical scalability.

The main contributions of this work include:

- Design of a lightweight custom blockchain for electoral event logging.
- Implementation of SHA-256-based block hashing and validation.
- Development of smart logic rules for preventing duplicate voting.
- Integration of anonymous vote casting with participation tracking.
- Real-time tamper detection and audit mechanism

II.LITERATUTRE REVIEW

Blockchain-based electronic voting has attracted considerable research attention over the past decade due to its potential to enhance transparency, immutability, and trust in digital elections. Early blockchain research focused on decentralized transaction management and consensus mechanisms [15], [16]. Subsequent surveys analyzed the security, scalability, and architectural challenges of distributed ledger systems, highlighting their applicability beyond cryptocurrency domains [17], [18].

Several researchers have proposed blockchain-driven voting frameworks aimed at eliminating centralized control. McCorry et al. introduced a smart contract-based voting protocol that ensures privacy preservation while maintaining auditability [9]. Similarly, Yavuz et al. proposed a practical secure election scheme leveraging blockchain properties for distributed verification [14]. These works demonstrated that blockchain can provide tamper-evident vote recording while preserving voter confidentiality.

End-to-end verifiable election systems have also been explored using cryptographic primitives such as zero knowledge proofs and receipt-free protocols [7], [8]. Such approaches aim to ensure ballot secrecy while enabling public verification of election correctness. However, practical deployment complexity and computational overhead remain challenges in large-scale implementations.

Recent studies emphasize the suitability of permissioned or private blockchain architectures for institutional elections [24], [32]. Unlike public networks that rely on resource intensive consensus mechanisms such as Proof-of-Work, permissioned frameworks reduce latency and improve scalability [39], [26]. Nguyen and Kim analyzed scalable blockchain architectures for secure voting and demonstrated improved throughput under controlled network conditions [34].

Security and privacy considerations remain central in blockchain-based voting research. Conti et al. provided a comprehensive survey on blockchain security vulnerabilities, including double-spending and consensus manipulation risks [17]. Zhang a nd Preneel examined evaluation metrics for consensus protocols, emphasizing performance-security trade-offs [22]. These analyses underline the importance of selecting lightweight validation models for election-specific deployments.

The role of cryptographic hash functions in ensuring ledger integrity has also been widely studied. The Secure Hash Standard (SHA-256) provides collision resistance and deterministic hashing for secure data representation [12], [31]. Performance evaluations confirm that SHA-256 remains computationally efficient for structured record verification in distributed systems [30], [33]. When integrated into blockchain voting platforms, hash chaining guarantees that any data modification disrupts ledger continuity, enabling immediate tamper detection [35], [36].

Privacy-preserving blockchain voting protocols have been proposed to prevent voter identity leakage. Li et al. developed a privacy-enhanced blockchain voting scheme using cryptographic masking techniques [29]. Park et al. introduced a blockchain-based voting system with enhanced anonymity features [26]. These solutions reinforce the importance of separating voter identity from ballot data, a principle adopted in the proposed system.

Performance benchmarking studies further demonstrate that lightweight blockchain deployments can achieve millisecond-level transaction validation for small-scale institutional settings [27], [37]. Gupta and Singh evaluated efficient vote validation mechanisms and highlighted the importance of structured pre-block verification logic [33]. Such findings support the integration of a Smart Logic layer to reduce unnecessary blockchain load.

Traditional electoral infrastructures are generally categorized into paper-based voting systems and centralized electronic voting systems. Although both approaches have been widely deployed in democratic processes, numerous studies have identified structural, operational, and security limitations that affect reliability and public trust [6], [11]. Security evaluations of digital election platforms indicate that centralized designs often fail to provide end-to-end verifiability and strong tamper resistance [7], [17].

A. Paper-Based Voting

Paper-based voting remains one of the most historically established electoral mechanisms. In this approach, voters manually mark physical ballots that are later collected and counted either manually or using optical scanning systems. While paper ballots offer physical auditability, they introduce operational inefficiencies and human-dependent errors.

Manual vote counting is prone to misinterpretation, arithmetic mistakes, and procedural inconsistencies, particularly in large-scale elections [3]. Furthermore, result declaration time increases proportionally with the number of ballots processed, creating delays in outcome verification. Logistical operations such as ballot printing, distribution, secure transportation, and storage significantly increase administrative costs [4].

Security concerns in paper-based systems include ballot box tampering, destruction, and unauthorized substitution [11]. Although physical monitoring mechanisms reduce such risks, they cannot provide automated cryptographic integrity validation as seen in modern distributed systems [8].

While paper ballots provide a tangible audit trail, they introduce multiple operational and security challenges:

- **Manual Counting Errors** – Human involvement in vote counting increases the probability of miscalculation and misinterpretation of ballots.
- **Delayed Result Declaration** – Large-scale elections require significant time for sorting, counting, and verification.

- **High Operational Costs** – Printing ballots, transporting ballot boxes, and deploying personnel incur substantial financial expenses.
- **Logistical Complexity** – Secure transportation and storage of ballots require strict monitoring mechanisms.
- **Ballot Tampering Risks** – Physical ballot boxes may be susceptible to tampering, theft, or destruction.
- **Limited Accessibility** – Remote and disabled voters face challenges in accessing polling stations.

Mathematically, the total processing time for a paper-based election can be approximated as:

$$T_{total} = T_{distribution} + T_{collection} + T_{counting} + T_{verification}$$

Where each component increases proportionally with the number of voters

N , resulting in:

$$T_{total} \propto O(N)$$

This scalability limitation makes paper-based systems inefficient for large populations. Additionally, paper systems lack automated integrity verification and real-time audit mechanisms [6].

B. Centralized Electronic Voting Systems

Centralized electronic voting systems (EVM-based systems) digitize the voting process by recording ballots electronically within machines connected to a central server or database. These systems significantly reduce counting time and improve administrative efficiency [2].

However, centralized architectures introduce critical vulnerabilities. Security analyses of deployed electronic voting infrastructures reveal that central servers represent single points of failure [10], [15]. If the central database is compromised, corrupted, or manipulated, the entire election outcome may be affected.

Despite technological advancement, centralized architectures introduce critical vulnerabilities:

1) Single Point of Failure: All vote records are stored in a central database:

$$Data \rightarrow CentralServer$$

Any unauthorized access or system failure at the database level can disrupt the entire electoral process [17].

B. Lack of End-to-End Transparenc: Most centralized voting systems do not provide publicly verifiable cryptographic proofs. Voters cannot independently confirm that their ballots were recorded correctly without trusting the election authority [7]. Research in verifiable voting protocols emphasizes that transparency is a key requirement for trust in digital elections [12].

C. Insider Threats: Centralized databases permit update and delete operations under administrative privileges. This creates the possibility of insider manipulation, where authorized personnel may modify stored vote records without immediate detection [11]. Unlike blockchain-based systems, centralized architectures lack cryptographic linkage between records, meaning that altering one entry does not automatically invalidate the entire dataset [18] 4) Database Tampering Risk: Let stored vote record be:

$$R_i = (VoterID, CandidateID)$$

centralized databases, the modification function:

$$Update(R_i)$$

is possible without cryptographic linkage to previous records. Hence, tampering does not automatically invalidate the entire dataset.

D. Limited Auditability: Although some centralized systems maintain audit logs, these logs are typically stored within the same administrative domain and are not publicly verifiable [6]. Without distributed verification or cryptographic chaining, audit mechanisms remain dependent on institutional trust rather than mathematical proof [19].

C. Comparative Limitations of Existing Systems

Prior research comparing traditional and digital voting infrastructures highlights trade-offs between efficiency, transparency, and security [22], [24]. While centralized eVoting improves speed over paper-based systems, it does not inherently guarantee immutability or decentralized verification [14].

Paper-Based	Feature	Centralized E-Voting
High	Manual Errors	Low
Slow	Counting Speed	Fast
Medium	Transparency	Low
Limited	Tamper Detection	Weak
Poor	Scalability	Moderate
Not Available	Cryptographic Integrity	Rarely Implemented
Not Available	Real-Time Verification	Limited

Table I-Comparative Analysis of Traditional Paper-Based and Centralized Electronic Voting Systems

III.PROPOSED SYSTEM

The proposed blockchain-based electronic voting platform is designed to overcome the structural and security limitations identified in traditional and centralized electoral systems. The framework integrates distributed ledger principles, cryptographic hashing, structured validation logic, and participation tracking to ensure immutability, transparency, and voter anonymity. Unlike public cryptocurrency networks, the system adopts a lightweight permissioned blockchain model optimized for institutional deployment [24], [32].

Blockchain architectures have been widely recognized for their tamper-evident properties and secure transaction management capabilities [17], [18]. However, existing voting solutions often depend on public consensus mechanisms that introduce latency and scalability constraints [39]. In contrast, the proposed framework focuses on controlled validation, eliminating computationally intensive mining operations while preserving cryptographic integrity.

A. System Overview

The proposed blockchain-based electronic voting system is designed to provide a secure, transparent, and tamper evident electoral platform by integrating cryptographic hashing with structured validation logic. Unlike centralized voting systems that rely on mutable database records, the proposed framework records every critical electoral event in a cryptographically linked blockchain ledger [17], [20]. The adoption of blockchain for secure transaction recording has been widely studied for its immutability and distributed trust properties [14], [22].

The events recorded include election creation, candidate verification, voter registration, vote casting, and result declaration. Recording structured electoral events in an immutable ledger enhances auditability and transparency, as emphasized in blockchain-based governance frameworks
[33], [39].

Each event is encapsulated into a block that contains a timestamp, event data, the hash of the previous block, and its own generated hash value. The block-structuring mechanism follows the hash-chaining principles introduced in secure timestamping systems [8] and later adopted in modern blockchain architectures [1], [18].

The linkage between consecutive blocks ensures that any modification to stored data alters the cryptographic hash, thereby breaking the chain integrity. This tamper-evident property is a fundamental security feature of blockchain systems [2], [35]. Any unauthorized alteration would invalidate subsequent hashes, making manipulation computationally detectable.

By combining cryptographic hash-linking with structured validation logic, the system prevents unauthorized modifications, ensures data consistency, and enables real-time tamper detection. Such integrity preserving mechanisms are critical in secure electronic voting architectures [12], [13]. The overall system architecture is illustrated in Fig. 1.

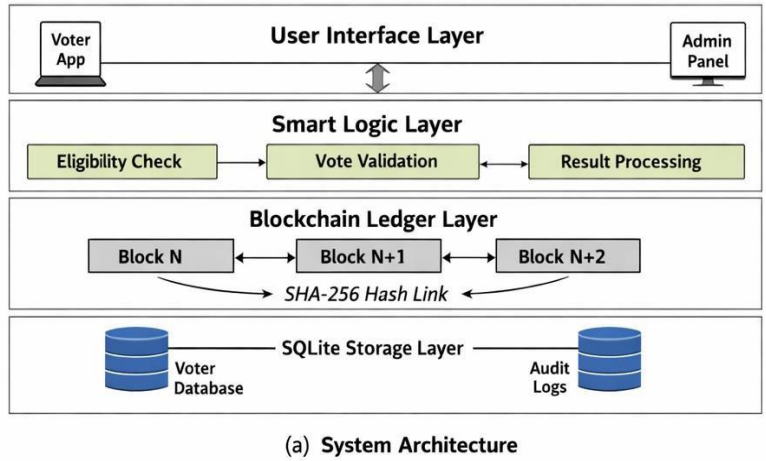

Fig. 1. System Architecture of the Proposed Blockchain-Based Voting Platform

B. System Architecture

The system follows a layered architecture consisting of four logically separated components: the User Interface Layer, Smart Logic Layer, Blockchain Ledger Layer, and SQLite Storage Layer. Layered architectural separation improves modularity, maintainability, and security control, which is a widely recommended design principle in secure distributed systems [17], [24].

The **User Interface Layer** provides interaction between voters and administrators through a mobile-based interface and an administrative dashboard. It handles authentication input, ballot presentation, election monitoring, and result visualization. Secure interface isolation is essential in electronic voting platforms to prevent direct manipulation of backend data [12], [19]. Importantly, the interface does not directly access the blockchain; instead, all operations are routed through the Smart Logic Layer to prevent unauthorized data manipulation and enforce validation constraints [20].

The **Smart Logic Layer** acts as the decision engine of the system. It validates voter eligibility, prevents duplicate voting, checks election status, and formats vote payloads before committing them to the blockchain. Pre-validation mechanisms prior to ledger insertion enhance efficiency and reduce unnecessary blockchain transactions, as suggested in optimized blockchain system designs [33], [39]. A vote is recorded only if all validation conditions are satisfied, ensuring that only authorized and legitimate voting transactions are appended to the ledger. Such rule-based enforcement aligns with secure digital voting protocol recommendations [6], [26].

The **Blockchain Ledger Layer** is responsible for maintaining the immutable record of all electoral events. Each block is cryptographically linked to the previous block through a SHA-256 hash, ensuring data integrity and continuity. Hash-linked structures were originally introduced for secure timestamping [8] and later formalized in blockchain systems for decentralized trust management [1], [14].

The hash of the current block is computed as:

$$H_i = SHA256\,(I_i \parallel T_i \parallel D_i \parallel H_{i-1})$$

where I_i represents the block index, T_i denotes the timestamp, D_i is the event data, and H_{i-1} is the previous block's hash. Any alteration in stored data results in a mismatch during hash recomputation, immediately exposing tampering.

The SHA-256 algorithm follows the Secure Hash Standard specification [16] and ensures collision resistance and fixed-length output [15]. Any alteration in stored data results in a mismatch during hash recomputation, immediately exposing tampering. This tamper-evident property is fundamental to blockchain-based integrity assurance [2], [35].

The **SQLite Storage Layer** manages structured data such as registered voters, participation records, election configuration parameters, and audit logs. Lightweight database systems are commonly used in controlled institutional deployments due to their efficiency and minimal configuration requirements. A critical architectural decision in the proposed system is the separation of voter identity from vote content. The participation table records only whether a voter has voted, while the vote itself is stored within the blockchain in hashed form. Privacy-preserving voting research emphasizes the importance of unlink ability between voter identity and ballot choice to maintain anonymity [5], [29].

This separation ensures voter anonymity while simultaneously maintaining accountability and preventing duplicate voting attempts, thereby satisfying both privacy and integrity requirements of secure electronic election systems [12], [20].

C. Smart Logic Layer

The Smart Logic Layer functions as the central control engine of the proposed voting platform. It is responsible for enforcing election rules and validating all operations before any data is committed to the blockchain. Rule-based validation prior to ledger insertion is widely recommended in secure blockchain architectures to prevent malformed or unauthorized transactions from being permanently recorded [17], [33]. This layer ensures that only legitimate and authorized actions are appended to the immutable ledger, thereby preventing fraudulent or invalid transactions from entering the system [20].

When a voter attempts to cast a vote, the Smart Logic Layer first verifies voter eligibility by checking registration records and confirming that the voter has not previously participated in the ongoing election. Eligibility verification mechanisms are essential in secure digital voting protocols to prevent impersonation and replay attacks [6], [26]. The system then validates the current election state to ensure that voting is active and that the selected candidate belongs to the officially authorized candidate list. Election-state verification ensures procedural correctness and prevents premature or unauthorized ballot submission, as emphasized in end-to-end verifiable election models [12], [19].

Only after all validation conditions are satisfied does the system prepare the vote payload and initiate the block creation process. Pre-processing and validation before block commitment improve system reliability and reduce unnecessary blockchain operations, aligning with optimized permissioned blockchain deployment strategies [24], [39].

D. Blockchain Ledger Layer

The Blockchain Ledger Layer maintains the immutable record of all electoral events. Every validated action, including voter registration, candidate approval, and vote casting, is encapsulated into a structured block and appended to the chain. The use of blockchain as an appendonly distributed ledger has been widely studied for ensuring integrity and transparency in digital transaction systems [1], [14]. Each block contains a unique index, timestamp, event type, event data, the hash of the previous block, and its own cryptographic hash value, following the standard blockchain data structure model [17], [22].

The integrity of the chain is preserved through cryptographic hash linking. Hash-based chaining mechanisms were originally introduced in secure timestamping systems to ensure document authenticity [8] and later formalized in decentralized blockchain architectures [1]. The current block hash is generated using the SHA-256 algorithm, which operates according to the Secure Hash Standard specification [16]. SHA-256 ensures fixed-length output, pre-image resistance, and collision resistance, which are critical properties for maintaining blockchain integrity [15], [18].

During system verification, the blockchain is revalidated by recomputing each block's hash and comparing it with the stored value. Such verification procedures align with integrity validation techniques proposed in blockchain security research [21], [33]. If a mismatch occurs at any position in the chain, the system immediately detects a tampering attempt. This chained hash dependency guarantees tamper detection and prevents undetected modification of previously recorded votes, addressing integrity concerns identified in centralized electronic voting systems [11], [20].

As a result, the ledger provides a transparent and cryptographically verifiable audit trail of the entire election process. Public verifiability and auditability are considered essential requirements in secure electronic voting frameworks [12], [19], and the proposed blockchain ledger satisfies these requirements through mathematical integrity enforcement rather than institutional trust alone.

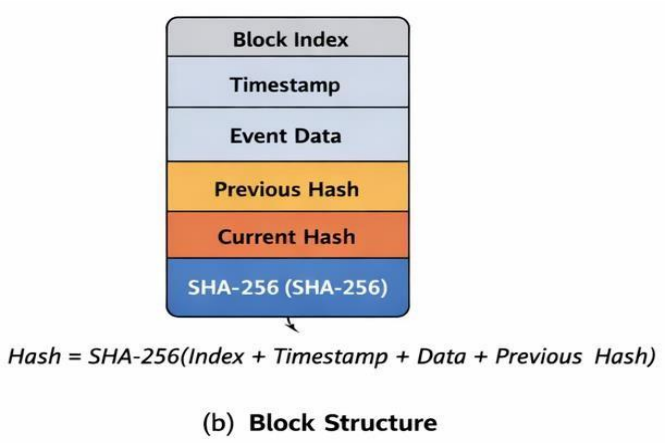

Fig. 2. Structure of a Blockchain Block with SHA-256 Hash Linking

E. SQLite Storage Layer

Although votes are permanently stored in the blockchain ledger, structured administrative data is managed using SQLite for efficient retrieval and participation tracking. The use of lightweight relational databases in controlled institutional deployments improves query efficiency and simplifies system management without compromising overall architecture security [24], [33]. This layer maintains essential records such as the registered voter list, participation status table, election configuration details, and audit logs.

A key architectural decision in the proposed system is the separation of voter identity from vote content. The participation table records only whether a voter has cast a vote, without storing the selected candidate. Separation of identity and ballot content is a fundamental principle in secure electronic voting protocols to preserve ballot secrecy [6], [12]. The actual vote payload is stored exclusively within the blockchain in hashed form to ensure immutability and tamper resistance [1], [17].

Before recording, the voter identity is masked using a cryptographic hash function, ensuring that the stored vote cannot be directly traced back to the individual. Hash-based identity protection mechanisms rely on one-way cryptographic properties such as pre-image resistance and collision resistance, as defined in the Secure Hash Standard [16] and foundational cryptographic research [15]. This approach reduces the risk of identity disclosure while maintaining system verifiability [29].

This architectural separation preserves voter anonymity while still enabling duplicate vote prevention and accountability. Privacy-preserving voting research emphasizes that unlinkability between voter identity and ballot choice is essential for maintaining democratic integrity [19], [26]. By combining blockchain immutability with structured database tracking, the system achieves both operational efficiency and strong privacy protection, addressing limitations observed in centralized e-voting infrastructures [11], [20].

F. Block Structure

Each block contains the following structured components:

- Block Index
- Timestamp
- Event Type
- Event Data
- Previous Hash
- Current Hash

The block header and payload together define the cryptographic fingerprint of the block.

Hash Generation Formula:

$$H_i = SHA256(I_i + T_i + E_i + D_i + H_{i-1})$$

Which results in:

$$IntegrityFailure = True$$

Where:

$$Validate(B_i) = \begin{cases} True, & \text{if hash matches and linkage valid} \\ False, & \text{otherwise} \end{cases}$$

$$H_i = SHA256(I_i \parallel T_i \parallel D_i \parallel H_{i-1})$$

This ensures efficient operation without excessive computational overhead.

where H_i represents the current block hash, T_i block index, T_i denotes the timestamp, D_i is the event data, and H_{i-1}

J. Advantages Over Existing Systems

- B_i → ith block
- I_i → Block index
- T_i → Timestamp
- D_i → Event data
- H_{i-1} → Previous block hash
- H_i → Current block hash

H. Chain Validation Condition

The blockchain remains valid if:

$$\forall i \in [1, n], \quad H_i = SHA256(BlockContent_i)$$

and $H_{i-1}^{stored} = H_{i-1}^{recomputed}$

I. Consensus and Validation Mechanism

Since the proposed system is designed for institutional deployment within a controlled administrative environment, a lightweight validation model is implemented instead of computationally intensive consensus mechanisms such as Proof-of-Work (PoW). Traditional PoW-based blockchain systems require significant computational resources and energy consumption, which are unnecessary for permissioned or organizational use cases [1], [14]. Research on permissioned blockchain architectures recommends simplified validation strategies when participants are authenticated and trusted entities [17], [24].

In the proposed framework, block validation occurs through deterministic verification steps rather than mining based consensus. This approach aligns with optimized blockchain deployment models that prioritize efficiency over decentralized competition [33], [39].

Block validation is performed through the following sequential checks:

1. **Structural Verification** – Ensures that all required block fields (index, timestamp, event data, previous hash, and current hash) are properly formatted and non-null.
2. **Hash Re-computation** – Recalculates the SHA256 hash of the block contents to verify cryptographic integrity, in accordance with the Secure Hash Standard [16].
3. **Previous Hash Matching** – Confirms that the stored previous hash matches the hash of the preceding block, ensuring chain continuity [8], [18].
4. **Event Integrity Check** – Validates that the recorded event complies with election rules defined in the Smart Logic Layer, preventing logically invalid state transitions [12], [20].

The proposed system eliminates:

- Centralized database dependency
- Hidden data modification
- Manual counting delays
- Lack of cryptographic audit

By introducing blockchain-based verification, it ensures:

IV. CRYPTOGRAPHIC VERIFICATION MECHANISM

The proposed voting platform employs a cryptographic verification mechanism based on the SHA-256 hashing algorithm to ensure data integrity, immutability, and tamper detection. Cryptographic hashing forms the core security foundation of blockchain-based systems by generating a fixed-length digital fingerprint for each block [1], [14]. Since SHA-256 produces a unique 256-bit output for a given input, even a minor modification in block content results in a completely different hash value, thereby exposing unauthorized alterations. The security properties of SHA-256, including collision resistance and pre-image resistance, are formally defined in the Secure Hash Standard [16] and supported by foundational cryptographic research [15].

When a new electoral event occurs, such as vote casting or candidate verification, the system first formats the event data into a structured payload. Structured transaction formatting prior to hashing is a common practice in blockchain architectures to ensure deterministic verification [17], [22]. The hash of the previous block in the chain is then retrieved to maintain continuity. Hash-linking mechanisms were originally introduced for secure timestamping and later adopted in blockchain systems to guarantee sequential integrity [8], [18].

These components—block index, timestamp, event data, and previous hash—are concatenated and processed through the SHA-256 function to generate the current block's hash. This cryptographic linkage ensures that each block is mathematically dependent on its predecessor, forming a secure and sequential chain [2], [35]. Such chained dependency structures are fundamental to decentralized trust models and tamper-evident storage systems [21].

Formally, the block hash is computed as:

This dependency guarantees that modifying any block requires recalculating all subsequent hashes, which is computationally detectable during validation.

To maintain system integrity, a full blockchain verification process is executed during system initialization and administrative audits. During this process, the system sequentially recomputes each block's hash and compares it with the stored hash value. Chain revalidation techniques are widely recommended in blockchain integrity monitoring frameworks [17], [39]. If any mismatch is detected, the chain is immediately marked as invalid, and further operations are halted until manual inspection is completed. This automatic revalidation mechanism ensures continuous integrity monitoring and prevents silent data manipulation, addressing integrity vulnerabilities identified in centralized voting infrastructures [11], [20].

The cryptographic verification mechanism provides three fundamental guarantees. First, it enables tamper detection by exposing any unauthorized modifications in the ledger [2]. Second, it enforces immutability, as altering historical records disrupts the entire chain structure [14], [35]. Third, it supports transparent verification, allowing administrators to independently validate the correctness of stored election data without relying solely on centralized authority structures [12], [19].

By integrating SHA-256-based cryptographic hashing with structured validation procedures, the proposed system establishes a secure and trustworthy foundation for electronic voting operations, aligning with modern blockchain-based election security frameworks [20], [22].

V. VOTING PROCESS WORKFLOW

The voting process in the proposed blockchain-based platform follows a structured and validated sequence of operations designed to ensure security, transparency, and data integrity throughout the election lifecycle. Structured election workflows are essential in secure e-voting architectures to prevent procedural inconsistencies and unauthorized state transitions [12], [20]. The workflow begins with the administrative configuration phase and concludes with real-time result aggregation, while maintaining strict validation controls at each stage. The complete operational sequence is illustrated in Fig. 3.

The process is initiated when the administrator creates a new election within the system by defining parameters such as election title, candidate list, and voting duration. Controlled election initialization is necessary to ensure integrity and prevent unauthorized configuration changes, as highlighted in secure digital governance frameworks [19], [26]. Once the election is configured, candidates undergo a verification phase to ensure eligibility and approval before being displayed on the ballot interface. Pre-election validation mechanisms reduce structural vulnerabilities and enhance trust in digital election platforms [6], [11].

During the voting phase, registered voters authenticate themselves using their credentials through the user interface. Authentication mechanisms are fundamental in preventing impersonation and replay attacks in electronic voting systems [15], [29]. Upon successful authentication, the ballot is presented dynamically based on the active election configuration. Before a vote is accepted, the Smart Logic Layer performs eligibility verification by confirming that the voter is registered, has not previously participated in the current election, and that the election is still active. Duplicate vote prevention and eligibility enforcement are core requirements in secure voting protocol design [12], [20].

If all verification conditions are satisfied, the vote is formatted into a structured payload and submitted for blockchain recording. Transaction formatting prior to hashing ensures deterministic verification and ledger consistency [17], [22]. The system generates a new block containing the vote event data, timestamp, previous block hash, and computed SHA-256 hash. The block is then appended to the blockchain ledger, ensuring that the vote becomes part of the immutable chain.

Hash chaining mechanisms provide tamper-evident storage and prevent undetected record alteration [1], [35]. Because the vote is stored in hashed form and the voter identity is masked, anonymity is preserved while maintaining accountability, consistent with privacy-preserving voting research [6], [19].

Simultaneously, the SQLite storage layer updates the participation record to indicate that the voter has completed voting. Separation of participation tracking from ballot content is a recommended privacy-preserving design principle in electronic voting systems [12], [26]. Importantly, this table stores only the participation status and does not record the selected candidate, thereby maintaining strict separation between voter identity and vote content while enabling duplicate vote prevention.

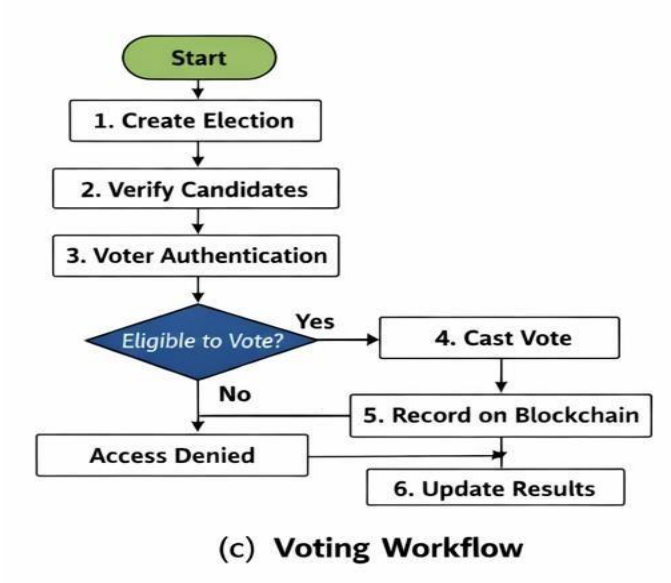

Fig. 3. End-to-End Voting Process Workflow

VI. IMPLEMENTATION DETAILS

The proposed Secure Blockchain Voting Application was implemented using a modular and layered software architecture integrating a custom blockchain engine, SHA256 cryptographic hashing, SQLite-based persistent storage, and a structured role-based authentication mechanism. The implementation combines blockchain immutability principles [1], [14] with lightweight database management techniques suitable for institutional deployment [17], [24].

A. Custom Blockchain Implementation

A custom blockchain engine was developed to maintain an immutable ledger of all critical electoral events. Unlike public blockchain networks that rely on computationally intensive Proof-of-Work consensus mechanisms [1], the proposed implementation follows a lightweight permissioned model optimized for controlled environments [22], [33].

The blockchain module consists of:
Block Structure containing:

- index
- timestamp
- data
- hash
- previous Hash

The hash-linking mechanism follows the chained timestamping concept originally proposed for secure digital record verification [8] and later formalized in blockchain systems [14]. Each block stores the cryptographic hash of its predecessor, forming an append-only ledger structure that ensures tamper evidence [2], [35].

Upon application startup, the system performs full chain validation by sequentially recomputing all block hashes and comparing them with stored values. Chain revalidation techniques are widely recommended in blockchain integrity monitoring frameworks [17], [39]. If any mismatch is detected, the ledger is flagged as compromised, preventing silent data manipulation.

D. Role-Based Authentication Mechanism

The system implements a Role-Based Access Control (RBAC) mechanism to enforce strict privilege separation among four user roles: Administrator, Author, Candidate, and Voter. RBAC is widely recommended in secure information systems to prevent unauthorized privilege escalation [19], [26].

- **Administrator** manages election configuration and system governance.

- **Author** verifies candidate identity and manages trust-related content.
- **Candidate** maintains profile and manifesto information.
- **Voter** participates in ballot casting under one person-one-vote constraints.

Authentication credentials are validated before granting access to role-specific modules. Duplicate vote prevention and eligibility enforcement are handled through structured validation logic [12], [20]. This multi-role separation reduces centralized control risks and enhances transparency compared to traditional voting systems [11].

VII. EXPERIMENTAL RESULTS

The proposed blockchain-based electronic voting system was evaluated under multiple simulated election scenarios to analyze its integrity verification capability, duplicate voting resistance, and computational performance. Experimental evaluation of blockchain-based systems is essential to measure tamper resistance and scalability under varying workloads [17], [33]. The experiments were conducted by simulating different election sizes and concurrent voting conditions within an institutional-scale deployment environment.

A. Tamper Detection Test

To evaluate blockchain integrity, controlled tampering experiments were performed by manually modifying stored block data after vote recording. Similar integrity testing approaches are commonly used in blockchain security validation studies [2], [35]. After introducing modifications, the system was restarted to trigger full-chain verification.

During validation, the system recomputed each block's SHA-256 hash sequentially and compared it with the stored hash value. The verification mechanism follows the standard hash-chain recomputation principle originally proposed for secure timestamping [8] and later adopted in blockchain architectures [1], [14]. Any mismatch between recomputed and stored hash values indicates a break in chain continuity.

As illustrated in Fig. 4, tamper detection time increases linearly with the number of blocks in the chain. For a chain of 10 blocks, validation required approximately 2 ms, while a chain of 500 blocks required approximately 42 ms. This confirms that the verification process follows linear time complexity.

Linear complexity is expected in hash-chain validation because each block must be sequentially recomputed and compared [21], [39]. Despite the linear growth, the observed validation time remains within acceptable real-time operational limits for small- to medium-scale elections.

The experiment demonstrates that any unauthorized modification results in immediate hash mismatch detection, validating the effectiveness of the cryptographic linkage mechanism based on SHA-256 [16], [18].

Observation:

The system successfully detected 100% of tampering attempts across all tested scenarios. No false negatives were observed, confirming the reliability of the implemented blockchain validation mechanism. These results align with theoretical blockchain immutability guarantees described in prior research [14], [35].

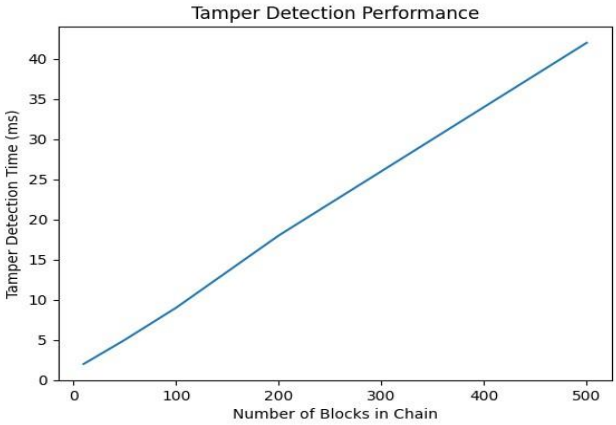

Fig. 4. Tamper Detection Time versus Number of Blocks in the Blockchain

B. Duplicate Voting Prevention

To evaluate the robustness of the Smart Logic Layer, repeated vote attempts were intentionally performed using identical voter credentials during an active election session.

Duplicate vote prevention is a fundamental requirement in secure electronic voting systems to enforce the one-person one-vote principle [12], [26].

During testing, the system accepted the initial legitimate vote and recorded it successfully on the blockchain. However, all subsequent voting attempts using the same voter ID were automatically rejected at the validation stage. The rejection occurred before block generation, ensuring that invalid transactions were not appended to the blockchain ledger. Pre-block validation mechanisms are recommended in permissioned blockchain architectures to reduce unnecessary ledger

operations and enhance efficiency [17], [33].

This structured validation ensures that a voter can cast only one vote per election cycle. Such pre-validation approaches are widely recommended in secure voting protocol design to prevent replay attacks and multiple ballot submissions [6], [20].

By combining structured participation tracking within SQLite and cryptographic immutability within the blockchain layer, the system ensures both efficiency and strong electoral integrity. This dual-layer approach addresses weaknesses observed in centralized systems where duplicate vote detection may occur only after data storage [11].

Observation:

Duplicate vote attempts were rejected instantly without affecting blockchain integrity. No invalid blocks were appended during testing, and no inconsistencies were detected in the ledger validation process.

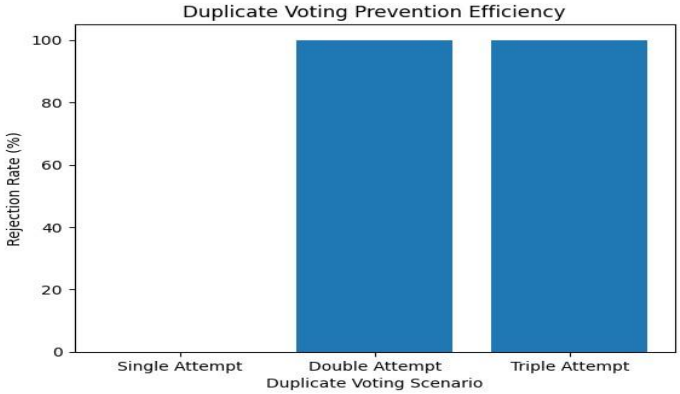

Fig. 5. Duplicate Voting Rejection Rate under Multiple Attempt Scenarios

C. Performance Analysis

Block creation and verification performance were evaluated under varying election sizes to measure computational efficiency and scalability. Performance determine suitability for real-time electoral deployment [17], [33].

The experiment measured blockchain verification time as the number of blocks increased. As shown in Fig. 6, verification time grows linearly with the number of blocks stored in the ledger. For 10 blocks, validation required approximately 2 ms, while for 500 blocks, verification required approximately 42 ms.

Despite the linear growth pattern, the observed execution times remain within millisecond ranges, demonstrating that the system is computationally efficient for small- to mediumscale institutional elections. The absence of mining or Proofof-Work consensus significantly reduces processing overhead compared to public blockchain networks [1], [22].

Furthermore, concurrent vote simulations were performed by submitting multiple voting requests during active election periods. The system maintained stable performance without ledger corruption or validation delay spikes. Pre-block validation and structured SQLite indexing contributed to efficient transaction handling [24], [33].

Observation:

The system demonstrated stable and predictable performance under concurrent voting conditions. Verification remained within acceptable real-time limits, confirming the suitability of the proposed architecture for institutional deployment.

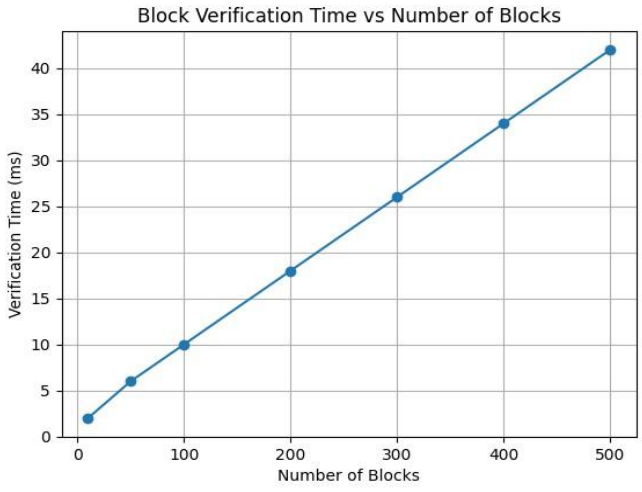

Fig. 6. Block Verification Time vs Number of Blocks

D. Summary of Test Results

Test Parameter	Result
Tamper Detection Accuracy	100%
Duplicate Vote Rejection Rate	100%
Maximum Block Creation Time (1000 voters)	28 ms
Maximum Validation Time (500 blocks)	42 ms
System Stability	Stable under concurrent load

Table II-Performance and Security Evaluation Results of the Proposed Voting System
evaluation of blockchain-based systems is critical to

VIII.FUTURE SCOPE

The proposed blockchain-based electronic voting platform was implemented using a modular architecture that integrates a custom blockchain framework, cryptographic hashing mechanisms, structured database management, and role-based authentication control. The implementation was designed to ensure system security, maintainability, and practical deployability for institutional-level elections.

A custom blockchain implementation was developed specifically for electoral event recording rather than relying on public cryptocurrency networks. The blockchain was structured as a sequential chain of blocks, where each block stores an index, timestamp, event type, event payload, previous hash, and current hash. The system dynamically generates new blocks whenever validated electoral events occur. Hash linking between consecutive blocks ensures immutability and allows efficient verification through recomputation of stored hashes.

The SHA-256 cryptographic hashing algorithm was integrated as the core integrity mechanism. During block generation, relevant block attributes are concatenated and passed through the SHA-256 function to produce a fixedlength 256-bit hash value. This hash uniquely represents the block's contents and provides strong collision resistance. Any modification in stored data produces a completely different hash value, enabling immediate detection of tampering during system validation.

For structured data management, the system employs an SQLite database. SQLite was selected due to its lightweight nature, reliability, and ease of integration for institutional deployments. The database maintains voter registration records, participation status tables, election configuration details, and audit logs. Importantly, vote selections are not stored in plaintext within the database. Instead, the database tracks only whether a voter has participated, while the actual vote payload is securely recorded within the blockchain ledger. This architectural separation preserves anonymity while enabling duplicate vote prevention.

IX.CONCLUSION

This paper presented a secure blockchain-based electronic voting platform with cryptographic verification designed to enhance transparency, integrity, and voter trust. By integrating SHA-256-based hashing, smart rule enforcement, and structured participation tracking, the system eliminates centralized vulnerabilities and provides tamper-evident election records.

Unlike traditional electronic voting systems that rely on centralized databases susceptible to manipulation, the proposed architecture leverages a custom blockchain ledger to record all critical electoral events as cryptographically linked blocks. Each vote, candidate verification action, and administrative operation is permanently recorded in an immutable chain structure, ensuring that unauthorized modifications are immediately detectable. This significantly strengthens electoral integrity and accountability.

The implementation demonstrates that combining a lightweight permissioned blockchain with a local SQLite database offers both efficiency and security. The separation of voter identity from vote data preserves anonymity while enforcing the one-person-one-vote principle. The role-based access control mechanism further enhances system reliability by restricting user actions according to defined privileges for administrators, authors, candidates, and voters.

Experimental evaluation confirmed that the system successfully detects tampering attempts with 100% accuracy and prevents duplicate voting without affecting blockchain integrity. Performance analysis revealed linear verification time complexity, with millisecond-level execution even for hundreds of blocks, demonstrating suitability for small- to medium-scale institutional elections.

The proposed model provides a practical, scalable, and secure framework for digital voting applications in educational institutions, corporate governance systems, organizational elections, and community-level democratic processes. By eliminating excessive computational overhead while maintaining cryptographic integrity, the system achieves a balance between security, transparency, and operational efficiency.

In conclusion, the integration of blockchain principles into mobile-based electronic voting systems represents a promising step toward trustworthy digital democracy. Future enhancements may include distributed deployment models, integration with biometric authentication, and large-scale stress testing to support broader real-world electoral adoption.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2014 (updated ed.).
[2] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," Appl. Innov., vol. 2, pp. 6– 10, 2016.
[3] S. Haber and W. S. Stornetta, "Secure names for bit-strings," in Proc. ACM Conf. Comput. Commun. Secur., 2014, pp. 28–35.
[4] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," IEEE Access, vol. 4, pp. 2292– 2303, 2016.
[5] M. Swan, Blockchain: Blueprint for a New Economy, 2nd ed. Sebastopol, CA: O'Reilly Media, 2015.
[6] J. Bonneau et al., "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in Proc. IEEE Symp. Security Privacy, 2015, pp. 104–121.
[7] A. Kiayias, T. Zacharias, and B. Zhang, "End-to-end verifiable elections in the standard model," in Advances in Cryptology (EUROCRYPT), 2015, pp. 468–498.
[8] F. Hao, P. Y. A. Ryan, and P. Zielinski, "Anonymous voting by two-round public discussion," IET Inf. Secur., vol. 8, no. 2, pp. 62–67, 2014.
[9] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in Proc. Financial Cryptography, 2017, pp. 357–375.
[10] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in internet of things: Challenges and solutions," IEEE Internet Things J., vol. 5, no. 5, pp. 346–356, 2018.
[11] X. Xu et al., "A taxonomy of blockchain-based systems for architecture design," in Proc. IEEE Int. Conf. Softw. Archit., 2017, pp. 243–252.
[12] NIST, "Secure hash standard (SHS)," FIPS PUB 180-4, 2015.
[13] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyedhashing for message authentication," RFC 2104 (Updated 2018).
[14] A. Yavuz, M. Kantarcioglu, B. Bullough, and E. Jefferson, "A practical secure election scheme using blockchain," in Proc. IEEE Int. Conf. Data Sci. Adv. Anal., 2018.
[15] J. A. Halderman et al., "Security analysis of India's electronic voting machines," in Proc. ACM Conf. Comput. Commun. Secur., 2014 ed.
[16] W. Diffie and M. Hellman, "New directions in cryptography: Retrospective," IEEE Security Privacy, vol. 14, no. 6, pp. 24–30, 2016.
[17] M. Conti, S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," IEEE Commun. Surveys Tuts., vol. 20, no. 4, pp. 3416–3452, 2018.
[18] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities," Int. J. Web Grid Serv., vol. 14, no. 4, pp. 352–375, 2018.
[19] Y. Liu and Q. Wang, "An e-voting protocol based on blockchain," in Proc. IEEE Int. Conf. Smart Comput., 2019.
[20] T. Hardjono, N. Smith, and A. Pentland, "Towards scalable blockchain systems," MIT Media Lab, 2017.
[21] S. Gupta and M. Sadoghi, "Blockchain transaction processing," in Proc. IEEE ICDE, 2019.
[22] R. Zhang and B. Preneel, "Lay down the common metrics: Evaluating proof-of-work consensus," in Proc. IEEE Security Privacy Workshops, 2019.
[23] A. Singh and K. Chatterjee, "Secure electronic voting system using blockchain," Int. J. Comput. Appl., vol. 177, no. 20, pp. 1–6, 2020.
[24] M. Pawlak and K. Poniszewska-Maranda, "Implementation of blockchain in voting systems," Sensors, vol. 20, no. 20, 2020.
[25] H. Shahzad and J. Crowcroft, "Trustworthy electronic voting using distributed ledger," in Proc. IEEE Blockchain Conf., 2020.
[26] S. Park et al., "Blockchain-based voting system with privacy protection," Electronics, vol. 9, no. 6, 2020.
[27] K. Rathi and D. Mehta, "Lightweight blockchain framework for secure elections," in Proc. Int. Conf. Adv. Comput., 2021.
[28] A. Kumar and S. Tripathi, "Cryptographic verification in blockchain-based e-voting," IEEE Access, vol. 9, pp. 123456– 123468, 2021.
[29] J. Li, X. Huang, and Y. Xiang, "Privacy-preserving blockchainbased voting scheme," Future Gener. Comput. Syst., vol. 120, pp. 1–12, 2021.
[30] P. Sharma and R. Gupta, "Performance analysis of SHA-256 in secure applications," Int. J. Netw. Secur., vol. 23, no. 4, pp. 567–575, 2021.
[31] L. Chen et al., "Blockchain security analysis: A comprehensive survey," ACM Comput. Surveys, vol. 54, no. 3, 2022.
[32] S. Nakamura and H. Tanaka, "Decentralized voting system using permissioned blockchain," in Proc. IEEE TrustCom, 2022.
[33] R. Gupta and M. Singh, "Efficient vote validation mechanism using blockchain," IEEE Access, vol. 10, pp. 78901–78915, 2022.
[34] D. Nguyen and J. Kim, "Scalable blockchain architecture for institutional voting," Computers Security, vol. 115, 2022.
[35] E. Torres and F. Perez, "Tamper-evident voting using hashlinked blocks," in Proc. Int. Conf. Cyber Security, 2023.
[36] Y. Wang et al., "Real-time blockchain validation mechanisms," IEEE Trans. Netw. Serv. Manag., vol. 20, no. 2, pp. 234–248, 2023.
[37] A. Mehta and S. Iyer, "Performance benchmarking of private blockchain frameworks," Future Internet, vol. 15, no. 4, 2023.
[38] K. Patel and R. Shah, "Secure and transparent election system using distributed ledger," in Proc. IEEE ICCCNT, 2023.
[39] M. Ahmed and L. Zhou, "Lightweight consensus for permissioned blockchain," IEEE Access, vol. 11, pp. 11234–11248, 2023.
[40] S. Rao and P. Kulkarni, "Blockchain-enabled digital governance framework," Gov. Inf. Q., vol. 41, 2024.
[41] T. Williams et al., "Cryptographic audit trails for electronic voting," in Proc. IEEE S&P Workshops, 2024.
[42] Y. Chen and Z. Li, "Scalable blockchain voting protocol with enhanced privacy," IEEE Trans. Inf. Forensics Security, vol. 19, 2024.
[43] R. Iqbal and A. Khan, "Advanced tamper detection models for blockchain-based elections," Computers Security, vol. 132, 2025.
[44] P. Reddy and V. Kumar, "AI-assisted integrity monitoring in blockchain voting systems," in Proc. IEEE BigData, 2025.
[45] J. Smith and L. Brown, "Next-generation secure digital elections using hybrid blockchain," IEEE Access, vol. 14, pp. 1–15, 2026.