International Journal of Circuit, Computing and Networking

E-ISSN: 2707-5931
P-ISSN: 2707-5923
Impact Factor (RJIF): 5.64
Journal's Website
IJCCN 2025; 6(2): 102-106
Received: 12-08-2025
Accepted: 17-09-2025

Rafael Yukio Shiraishi
Department of Informatics, Federal Institute of Education, Science and Technology of São Paulo (IFSP), Bragança Paulista, SP, Brazil

Flavio Cezar Amate
Department of Informatics, Federal Institute of Education, Science and Technology of São Paulo (IFSP), Bragança Paulista, SP, Brazil

Corresponding Author:
Flavio Cezar Amate
Department of Informatics, Federal Institute of Education, Science and Technology of São Paulo (IFSP), Bragança Paulista, SP, Brazil

Decentralized electronic voting system using smart contracts on the TRON blockchain

Rafael Yukio Shiraishi and Flavio Cezar Amate

DOI: https://www.doi.org/10.33545/27075923.2025.v6.i2b.115

Abstract
This paper presents the design and implementation of a decentralized electronic voting system based on a hybrid architecture that integrates the TRON blockchain with off-chain authentication mechanisms. The proposed solution employs smart contracts written in Solidity to record votes in an immutable and publicly auditable manner, while a backend service implemented in Node.js and a MySQL database handles voter authentication and enforces voter uniqueness. To prevent duplicate voting and ensure auditability, cryptographic hash functions are used to bind voter credentials and election parameters to each vote without exposing sensitive data on-chain. Experimental results demonstrate that the system effectively mitigates common security threats, such as duplicate voting and unauthorized data manipulation, while maintaining low transaction costs and practical usability. The findings indicate that the proposed hybrid approach provides a secure, transparent, and cost-effective alternative for electronic voting systems in real-world scenarios.

Keywords: Blockchain, Electronic Voting, Smart Contracts, TRON, Security, Web3.0

1. Introduction

Electronic voting systems have been increasingly adopted in academic, organizational, and governmental contexts due to their potential to improve efficiency, reduce operational costs, and accelerate vote counting processes. However, traditional electronic voting solutions are commonly based on centralized architectures, which introduce critical challenges related to transparency, trust, auditability, and security. Centralized databases and servers represent single points of failure and are vulnerable to manipulation, unauthorized access, and data tampering, which can compromise the integrity of electoral processes [1, 2, 3, 4].

In recent years, blockchain technology has emerged as a promising alternative to mitigate these limitations. By providing decentralization, immutability, and public verifiability, blockchain-based systems enable the creation of transparent and tamper-resistant records without relying on trusted third parties [2, 3, 6]. These characteristics make blockchain particularly suitable for electronic voting applications, where trust, integrity, and auditability are essential requirements. Smart contracts further enhance this paradigm by allowing the automation of voting rules and vote counting logic in a deterministic and verifiable manner [3, 5, 7].

Several studies have explored blockchain-based electronic voting systems using platforms such as Ethereum and Solana, highlighting advantages related to transparency and security, as well as challenges involving scalability, transaction costs, and usability [2, 3, 8, 9]. High transaction fees and network congestion, commonly observed in some public blockchains, may limit the feasibility of large-scale voting systems. In this context, the TRON blockchain presents itself as an attractive alternative due to its high throughput, low transaction costs, and support for smart contracts written in Solidity [10]. TRON adopts a Delegated Proof of Stake (DPoS) consensus mechanism, which contributes to faster transaction confirmation times and reduced operational costs when compared to traditional Proof of Work-based networks [11].

Despite these advantages, the design of a secure and practical blockchain-based voting system still faces important challenges, particularly regarding voter authentication, prevention of duplicate voting, and the balance between decentralization and usability. Fully on-chain identity management may expose sensitive information or increase system complexity, while purely off-chain solutions may weaken auditability and trust [2, 12]. Therefore, hybrid architectures that combine blockchain features with traditional authentication mechanisms have gained attention as a pragmatic approach.

This paper presents the design and implementation of a decentralized electronic voting system based on a hybrid architecture that integrates the TRON blockchain with an off-chain authentication infrastructure. Smart contracts deployed on the TRON network are responsible for registering elections, recording votes in an immutable manner, and enabling transparent vote counting. A backend service, implemented using Node.js and a relational MySQL database, handles voter authentication and enforces voter uniqueness through token-based verification mechanisms. To prevent vote duplication and ensure auditability, cryptographic hash functions are employed to bind voter credentials and election parameters to each recorded vote [8, 12].

The main contributions of this work are threefold: (i) the proposal of a hybrid voting architecture that combines blockchain immutability with traditional authentication mechanisms; (ii) the implementation of secure smart contracts on the TRON blockchain to support decentralized voting processes; and (iii) an experimental evaluation demonstrating the system's resistance to common attack vectors such as duplicate voting and unauthorized vote manipulation.

2. Materials and methods

This section describes the materials, technologies, and methodological approach adopted in the design and implementation of the proposed decentralized electronic voting system. The system follows a hybrid architecture that combines blockchain-based components for transparency and immutability with off-chain services for authentication and access control.

2.1 System Architecture: The proposed voting system is based on a hybrid architecture composed of four main layers: (i) user interface, (ii) application backend, (iii) blockchain layer, and (iv) data persistence for authentication. This architectural design aims to balance decentralization, security, and usability, combining on-chain vote immutability with off-chain identity management. Figure 1 illustrates the overall system architecture and the interactions among its main components.

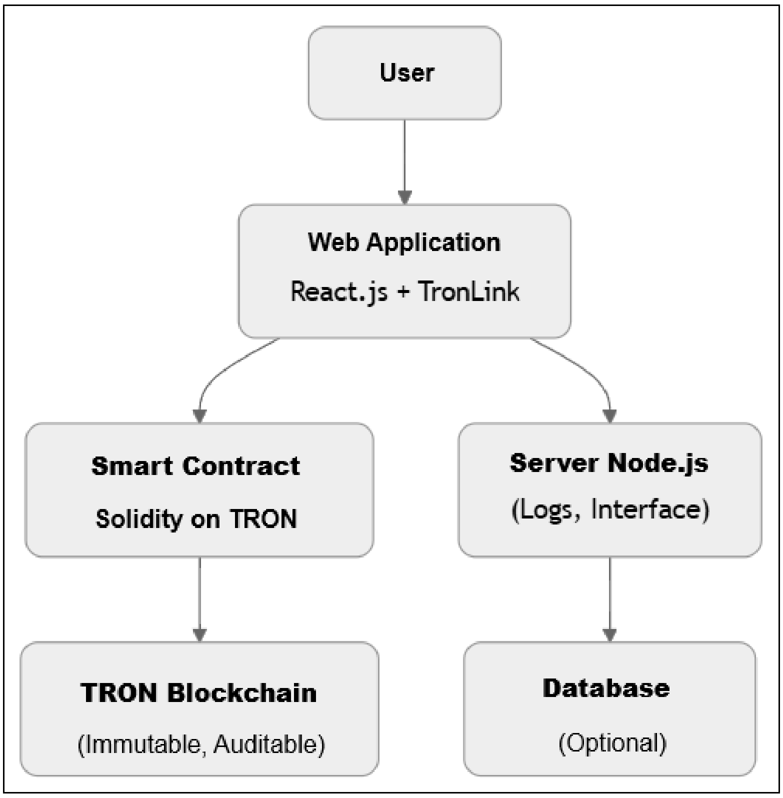

Fig 1: Hybrid on-chain/off-chain architecture integrating a web application, smart contracts, and the TRON blockchain.

The user interface was developed as a web-based application using ReactJS, allowing voters and administrators to interact with the system through a standard web browser. The frontend communicates with the TRON blockchain using the TronWeb JavaScript API and relies on the TronLink wallet extension to manage cryptographic keys and sign transactions [15]. This approach ensures that sensitive blockchain operations, such as vote submission, are executed directly by the user without exposing private keys to the application server.

The backend layer was implemented using Node.js and is responsible for voter authentication, token generation, and validation. This layer acts as a controlled gateway that verifies voter eligibility before allowing interactions with the blockchain. A MySQL relational database is used to store voter credentials and authentication states, ensuring voter uniqueness and preventing duplicate voting attempts.

The blockchain layer consists of smart contracts deployed on the TRON public network. These contracts handle election creation, vote registration, and vote counting in an immutable and transparent manner. Once a vote is recorded on-chain, it cannot be altered or removed, enabling public auditability of the election process.

This hybrid architecture allows the system to benefit from the transparency and immutability of blockchain technology while maintaining efficient and secure identity verification through traditional authentication mechanisms.

2.2 Smart Contract Design

Smart contracts were developed using the Solidity programming language and executed on the TRON Virtual Machine (TVM) [14]. The contracts define the core logic of the voting process and are responsible for maintaining the integrity of elections and votes.

Each election is represented by a structured data model containing metadata such as title, description, list of candidates, and voting status. The contract provides functions to create elections, register votes, and retrieve results. To ensure correctness and security, state-modifying functions include validation checks that verify election status, candidate validity, and voter eligibility.

Vote registration is implemented through a function that receives election identifiers and candidate selections. To prevent duplicate voting, the contract stores cryptographic hashes associated with each vote. Before accepting a new vote, the contract verifies whether a corresponding hash already exists, ensuring that each eligible voter can vote only once per election.

All read-only operations, such as retrieving election data and vote counts, are implemented as view functions that do not alter the contract state. This separation between read and write operations reduces the risk of unintended state changes and improves contract safety.

Once deployed, the smart contract address is publicly disclosed and serves as the single trusted reference for vote verification and auditing. Any observer can independently inspect the contract state and validate the election results using publicly available blockchain data.

2.3 Authentication and Backend Infrastructure

While the blockchain ensures immutability and transparency, voter authentication is handled off-chain to avoid exposing sensitive personal information. The backend infrastructure implements a token-based authentication mechanism that validates voter identity before allowing vote submission.

Voters authenticate using a unique identifier, password, and registered email address. Upon successful authentication, the backend generates a one-time verification token and sends it to the voter via email. The vote can only be confirmed after the token is validated, ensuring that the registered email address is under the control of the voter.

The MySQL database enforces voter uniqueness by associating each voter record with a unique email address. Authentication attempts are logged and verified before blockchain interaction is authorized. If a voter attempts to vote more than once, the backend prevents token reissuance and blocks further submissions.

This off-chain authentication strategy reduces on-chain complexity, improves system usability, and minimizes the risk of identity exposure on a public blockchain, while still enforcing strong access control policies.

2.4 Security Mechanisms

Multiple security mechanisms were incorporated into the system to mitigate common attack vectors associated with electronic voting systems.

To prevent duplicate voting, a dual-layer validation approach is employed. First, the backend verifies voter eligibility and token validity. Second, the smart contract checks whether a cryptographic hash associated with the voter and election has already been recorded on-chain. This layered defense significantly reduces the risk of replay or double-voting attacks.

Cryptographic hash functions are used to generate audit hashes that bind voter identifiers, election parameters, and timestamps. These hashes are stored on the blockchain instead of raw personal data, preserving voter privacy while enabling auditability. To ensure vote uniqueness and prevent duplicate voting, a cryptographic hash is generated for each vote using the Keccak-256 function, Eq.(1).

$$H_{vote} = \text{Keccak256}\,(ID_{voter} \parallel Email \parallel Title_{election} \parallel Date_{election}) \quad (1)$$

Access control risks at the smart contract level are mitigated by restricting state-changing functions to predefined workflows and publicly documented contract addresses. Any attempt to deploy unauthorized contracts is detectable, as only the official contract address is considered valid for election auditing.

Finally, all blockchain transactions are signed locally through the TronLink wallet, ensuring that private keys are never exposed to the application backend. This design choice protects voters against key leakage and unauthorized transaction signing.

2.5 Implementation Environment

The system was implemented and tested using a web-based frontend developed with ReactJS, integrated with the TRON blockchain through the TronWeb library and the TronLink wallet. The backend infrastructure was implemented using Node.js, while a MySQL relational database was employed for data persistence and voter authentication. The blockchain layer relies on the TRON public network, and all smart contract logic was developed using the Solidity programming language.

Functional and security tests were conducted to validate election creation, voter authentication, vote submission, prevention of duplicate voting, and result retrieval. These tests confirmed the correct integration of all system components and the effectiveness of the proposed security mechanisms.

3. Results and Discussion

This section presents the experimental results obtained from the implementation of the proposed decentralized voting system and discusses its security, integrity, and practical feasibility. The evaluation focuses on functional correctness, resistance to common attack vectors, and the effectiveness of the hybrid architecture.

3.1 Functional Validation

The system was evaluated through a series of functional tests covering the complete voting lifecycle, including election creation, voter authentication, vote submission, election closure, and result retrieval. All stages were successfully executed without inconsistencies or unexpected behavior.

Election creation transactions were correctly recorded on the TRON blockchain, generating immutable records containing election metadata and candidate lists. Once deployed, elections became publicly verifiable through the smart contract state, allowing any observer to inspect election parameters and verify their authenticity.

The voting process demonstrated stable integration between the frontend, backend, and blockchain layers. Votes submitted through the web interface were properly authenticated, signed locally via the TronLink wallet, and registered on-chain. After election closure, vote counting functions accurately reflected the total number of votes per candidate, confirming the correctness of the implemented smart contract logic.

These results demonstrate that the proposed system fulfills its functional requirements while maintaining consistency between off-chain and on-chain components.

3.2 Security Evaluation

Duplicate voting prevention was one of the primary security objectives. The implemented dual-layer validation approach proved effective, combining off-chain authentication controls with on-chain hash verification. Similar multi-layer approaches have been recommended in previous blockchain-based voting studies as a means to reduce replay and double-voting attacks [2, 8]. Table 1 summarizes the main security threats identified during system evaluation, along with the corresponding mitigation strategies and residual risk assessment.

Table 1: Security threats, mitigation strategies, and residual risk assessment of the proposed voting system.

Security Threat / Test Case	Mitigation Strategy	Residual Risk
Unauthorized election creation or closure	Only the officially published smart contract address is considered valid; authorized wallet list is publicly auditable	Low
Vote tampering or incorrect vote recording	Vote hashes are immutably stored on the blockchain and validated by the backend before confirmation	Low
Duplicate voting attempts	Dual-layer verification using backend authentication and on-chain hash validation	Low
Voter identity manipulation	Email-based token confirmation and unique database constraints prevent impersonation	Very Low
Post-vote data modification	Blockchain immutability guarantees that recorded vote hashes cannot be altered	Low
Unauthorized state changes through read functions	All read-only functions are implemented as view operations	Low
Unauthorized vote counting or result manipulation	Public smart contract enables transparent and verifiable vote counting	Low

Vote integrity and immutability were ensured by recording vote-related hashes on the blockchain. Once stored, these hashes could not be altered, providing strong guarantees against post-vote manipulation. This result is consistent with prior findings that highlight blockchain immutability as a core advantage over centralized voting systems [6, 12].

Unauthorized state modification risks were mitigated by restricting smart contract write operations and separating read-only functions from state-changing logic. This design aligns with best practices for secure smart contract development and minimizes common vulnerabilities reported in Solidity-based systems [13]. All identified threats resulted in low or very low residual risk, as summarized in Table 1.

3.3 Discussion of the Hybrid Architecture

The experimental results highlight the effectiveness of the hybrid architecture in balancing decentralization, security, and usability. Unlike fully on-chain voting systems, which often suffer from high complexity and privacy concerns, the proposed approach leverages off-chain authentication to manage voter identities securely.

From an architectural perspective, delegating identity verification to the backend reduces on-chain storage requirements and transaction costs. At the same time, the blockchain layer remains responsible for the most critical aspects of election integrity: vote recording, immutability, and public auditability. This separation of responsibilities enhances scalability and simplifies system maintenance.

The choice of the TRON blockchain proved advantageous in terms of performance and cost efficiency. Transaction confirmation times were suitable for real-time voting scenarios, and the low transaction fees make the solution viable for elections with a larger number of participants. Compared to blockchain platforms with higher gas costs, TRON offers a more accessible environment for decentralized voting applications.

However, the hybrid model also introduces certain trade-offs. The backend remains a semi-trusted component responsible for authentication, which means that complete decentralization is not achieved. Nevertheless, the immutability of on-chain vote records ensures that even if the backend is compromised, vote manipulation after submission remains infeasible. This trade-off is considered acceptable for practical deployments where usability and compliance requirements are critical.

3.4 Comparison with Related Work

Compared to existing blockchain-based voting systems, many solutions rely exclusively on fully on-chain identity management, which may expose sensitive information or increase system complexity [2, 9]. Other approaches remain fully centralized, limiting transparency and auditability.

The proposed system adopts a hybrid model that combines off-chain identity validation with on-chain vote immutability. Similar architectural trade-offs have been discussed in recent studies, which argue that hybrid approaches offer better usability and scalability for real-world deployments [8, 12]. Additionally, the use of the TRON blockchain addresses performance and cost limitations reported in systems deployed on more congested networks [10, 11].

4. Conclusions

This paper presented the design and implementation of a decentralized electronic voting system based on a hybrid architecture that integrates blockchain technology with traditional authentication mechanisms. By combining smart contracts deployed on the TRON blockchain with an off-chain backend for voter authentication, the proposed solution achieves a balance between transparency, security, and practical usability.

The results demonstrate that the use of blockchain as an immutable ledger for vote recording effectively enhances the integrity and auditability of the electoral process. The implementation of cryptographic hashing mechanisms and on-chain validation rules successfully prevents vote duplication and unauthorized state modifications. At the same time, the off-chain authentication layer ensures voter uniqueness and identity verification without exposing sensitive personal information on a public blockchain.

The choice of the TRON blockchain proved suitable for decentralized voting applications due to its low transaction costs, fast confirmation times, and compatibility with Solidity-based smart contracts. These characteristics make the proposed system viable for real-world scenarios where scalability and operational efficiency are critical factors.

Despite these advantages, the proposed architecture is not fully decentralized, as the authentication backend remains a semi-trusted component. However, this trade-off is considered acceptable in contexts where usability, privacy, and regulatory constraints must be addressed. Importantly, even in the presence of backend compromise, the immutability of blockchain records ensures that recorded votes cannot be altered or removed, preserving the integrity of election outcomes.

Future work will focus on enhancing decentralization by introducing on-chain access control mechanisms, improving privacy through advanced cryptographic techniques, and extending the system to support large-scale elections and more complex voting models. Additional evaluations involving stress testing, formal verification of smart contracts, and usability studies with larger user groups are also planned.

Overall, the proposed system demonstrates that a hybrid blockchain-based approach can provide a secure, transparent, and cost-effective alternative to traditional electronic voting systems, contributing to ongoing research on decentralized applications and trustworthy digital governance.

5. Acknowledgments

The authors would like to thank the Federal Institute of Education, Science, and Technology (IFSP) for the financial and institutional support provided through the Institutional Scientific Initiation Scholarship Program (PIBIFSP).

References

1. Ferreira JE, Pinto FGC, dos Santos SC. Estudo de mapeamento sistemático sobre as tendências e desafios do Blockchain [A systematic mapping study on blockchain trends and challenges]. Gestao.org. 2017;15(6):108-117.
2. Jafar U, Aziz MJA, Shukur Z. Blockchain for electronic voting system—review and open research challenges. Sensors. 2021;21(17):5874.
3. Zheng Z, Xie S, Dai H, Chen X, Wang H. An overview of blockchain technology: Architecture, consensus, and future trends. In: Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress); 2017 Jun; Honolulu, HI. IEEE; 2017. p. 557-564.
4. Crosby M, Pattanayak P, Verma S, Kalyanaraman V. Blockchain technology: Beyond bitcoin. Applied Innovation. 2016;2:6-10, 71.
5. Christidis K, Devetsikiotis M. Blockchains and smart contracts for the internet of things. IEEE Access. 2016;4:2292-2303.
6. Wu K, Ma Y, Huang G, Liu X. A first look at blockchain-based decentralized applications. Software Pract Exp. 2021;51(10):2033-2050.
7. Solidity Documentation. Solidity: Language Documentation. 2025.
8. Anitha V, Caro OJM, Sudharsan R, Yoganandan S, Vimal M. Transparent voting system using blockchain. Measurement: Sensors. 2023;25:100620.
9. Hassan HS, Hassan RF, Gbashi EK. E-voting system using Solana blockchain. In: Proceedings of the 4th International Conference on Current Research in Engineering and Science Applications (ICCRESA); 2022 Dec. IEEE; 2022. p. 147-153.
10. TRON Foundation. TRON whitepaper. 2023.
11. Li X, Wang X, Kong T, Zheng J, Luo M. From Bitcoin to Solana—innovating blockchain towards enterprise applications. In: International Conference on Blockchain. Cham: Springer; 2021. p. 74-100.
12. Ibrahim M. Design and development of a decentralized voting system using blockchain. Int J Commun Inf Technol. 2023;4(1):1-11. doi:10.33545/2707661X.2023.v4.i1a.53.
13. Tantikul P, Ngamsuriyaroj S. Exploring vulnerabilities in Solidity smart contract. In: Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP); 2020 Feb. p. 317-324.
14. TRON Foundation. TRON Virtual Machine (TVM) documentation. 2025.
15. TRON Foundation. TronWeb JavaScript API documentation. 2025.