iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VIII **Month of publication:** August 2026

DOI: https://doi.org/10.22214/ijraset.2026.82170

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VIII Aug 2026- Available at www.ijraset.com

A Blockchain and Machine Learning Framework for Secure E-Voting Systems with Intrusion Detection

Mrs. A. Asrin Mahmoodha[1], Dr. B. Aysha Banu[2], Mohammed Muhajir S[3], Mohamed Anas A[4], Mohamed Anas R[5]
[1]*Assistant Professor,* [2]*Professor and Head,* [3, 4, 5]*UG Students, Department of Information Technology, Mohamed Sathak Engineering College (Autonomous), Kilakarai-623806, Tamil Nadu, India*

Abstract: ***Voting is a central component of a country's political life cycle. Privacy, authentication, and integrity of citizens' votes are essential requirements of any electronic voting programme. To address these concerns, this paper proposes a hybrid e-voting system that integrates personal and public blockchain with a machine learning–based intrusion detection mechanism. The personal blockchain governs voter registration and vote casting, while the public blockchain stores the Merkle root hash for result integrity verification. An ML-based intrusion detection system monitors voting data centres and e-voting stations for anomalous behaviour. Homomorphic encryption and zero-knowledge proofs preserve voter anonymity. Experimental evaluation demonstrates that the proposed framework achieves an accuracy of 97.4%, precision of 96.8%, recall of 97.1%, and F1-score of 96.9% in detecting intrusion attempts. The system also reduces transaction latency by 34% compared to conventional blockchain voting systems. Results confirm that the framework delivers transparency, tamper-resistance, and strong security guarantees, making it a viable solution for modern democratic elections.***

Keywords: ***Blockchain, e-voting, machine learning, intrusion detection, Merkle hash tree, homomorphic encryption, zero-knowledge proof, smart contracts, cybersecurity.***

I. INTRODUCTION

Electronic voting (e-voting) has emerged as a promising alternative to conventional paper-based and electronic voting machine (EVM) approaches, offering remote accessibility and operational efficiency. However, existing e-voting deployments expose critical vulnerabilities including single points of failure, susceptibility to Distributed Denial-of-Service (DDoS) attacks, vote tampering, and inadequate voter authentication. High-profile elections have witnessed cyberattacks targeting centralised voting databases, eroding public confidence in the integrity of results. Blockchain technology offers a decentralised, immutable ledger that records transactions without reliance on a central authority. Its cryptographic properties — including hash chaining, digital signatures, and consensus mechanisms — make it inherently resistant to post-hoc data modification. Several researchers have proposed blockchain-based e-voting systems; however, few have simultaneously addressed the insider threat posed by compromised voting infrastructure nodes, which may relay fraudulent transactions that appear cryptographically valid. Machine learning (ML)–based intrusion detection systems (IDS) complement blockchain by monitoring data traffic patterns at voting data centres and e-voting stations. ML classifiers trained on network behaviour can flag anomalous access patterns in real-time, thereby preventing fraudulent ballots from entering the blockchain before they become immutable. The proposed system introduces a dual-blockchain architecture: a permissioned personal blockchain for voter registration and vote casting, and a public blockchain for storing Merkle root hashes and broadcasting results. This separation of concerns limits the attack surface while preserving public verifiability. Voter anonymity is preserved through homomorphic encryption and zero-knowledge proofs.

The main contributions of this paper are as follows:

1) A hybrid personal–public blockchain architecture is proposed for e-voting that separates registration from result publication.
2) An ML-based intrusion detection module is integrated to identify anomalous behaviour in voting infrastructure in real time.
3) Merkle hash tree–based vote integrity verification is implemented to detect any tampering with individual vote records.
4) Homomorphic encryption and zero-knowledge proofs are incorporated to preserve voter anonymity while enabling vote verification.
5) Experimental evaluation on a simulated election dataset demonstrates superior performance over state-of-the-art baselines in accuracy, latency, and scalability.

II. RELATED WORK

Ayed [1] presented one of the earliest conceptual frameworks for blockchain-based e-voting, arguing that the decentralised ledger can eliminate the need for a trusted third party. However, the work lacked implementation details and performance evaluation.

Sharma et al. [2] implemented a blockchain-based online voting system using Ethereum smart contracts. The system ensured transparency and vote immutability, but relied on public blockchain consensus (Proof-of-Work), which introduced high transaction latency unsuitable for large-scale national elections.

Kumar et al. [3] proposed an AI-driven voter authentication framework combining facial recognition and biometric verification. While the authentication layer was robust, the underlying vote-storage mechanism remained centralised, preserving the single-point-of-failure risk. Patil and Sharma [4] explored cloud and IoT-based smart voting, achieving improved remote accessibility. The work, however, did not address vote integrity beyond standard cloud-security measures and lacked a decentralised verification mechanism.

Brown and Lee [5] applied homomorphic encryption to online voting, enabling tallying without decrypting individual ballots. The approach preserved privacy but incurred significant computational overhead, and no intrusion detection mechanism was described.

Wilson and Carter [6] conducted a comparative study of existing online voting systems, highlighting the research gap: no existing system simultaneously addressed integrity, anonymity, scalability, and intrusion detection in a unified framework.

The present work fills this gap by combining a dual-blockchain architecture with an integrated ML-based IDS and Merkle-tree integrity verification, achieving a balance among security, privacy, and performance that existing solutions do not provide.

TABLE I
COMPARISON OF EXISTING E-VOTING SYSTEMS

Paper	Technology	Security Mechanism	Intrusion Detection	Limitation
Ayed [1]	Conceptual blockchain	Decentralisation	No	No implementation
Sharma [2]	Ethereum PoW	Smart contracts	No	High latency
Kumar [3]	AI + centralised DB	Biometric auth	Partial	Single point of failure
Patil [4]	Cloud + IoT	Cloud security	No	No integrity proof
Brown [5]	Homomorphic enc.	Encrypted tallying	No	High compute overhead
Proposed	Dual blockchain + ML	Merkle + HE + ZKP	Yes (ML IDS)	None identified

III. PROPOSED FRAMEWORK

A. *System Overview*

The proposed framework comprises five integrated modules: (1) Admin Module, (2) User/Voter Module, (3) Personal Blockchain Network, (4) Public Blockchain Network, and (5) ML-Based Intrusion Detection Module. Figure 1 presents the top-level system architecture. Administrators create election events, verify voter eligibility, and oversee the voting process through a dashboard. Voters authenticate using multi-factor credentials (password + biometric token) and cast ballots through an encrypted channel. Each ballot is encrypted, validated by the personal blockchain, and its Merkle root published to the public blockchain at the close of polling.

Fig. 1. Proposed System Architecture

B. Personal Blockchain Design

The personal blockchain is a permissioned Hyperledger Fabric network. Each node represents a registered voting station. Voter registration records and encrypted ballots are recorded as transactions. The consensus algorithm is Practical Byzantine Fault Tolerance (PBFT), ensuring finality within three communication rounds and tolerating up to f malicious nodes out of 3f + 1 total nodes.

Each block B_i is structured as:

$B_i = \{Header_i, Transaction\_List_i, Merkle\_Root_i\}$

where $Header_i$ contains the block index, timestamp, previous block hash, and nonce. The transaction list stores encrypted vote records, and $Merkle\_Root_i$ is computed from the SHA-256 hashes of individual transactions.

C. Public Blockchain Design

The public blockchain is an Ethereum-compatible network accessible to any observer. It stores only the Merkle root hashes of each polling station's personal blockchain and the aggregated election result. Smart contracts govern automatic result publication upon the closure of the voting window, without requiring administrator intervention, thereby eliminating result manipulation opportunities.

D. Intrusion Detection Module

The ML-based IDS is deployed at each e-voting station and central data centre. It monitors network traffic features including packet inter-arrival time, flow byte rate, connection duration, protocol type, and source/destination port entropy. A Random Forest classifier trained on the KDD Cup 1999 and UNSW-NB15 intrusion detection datasets classifies each observed flow as benign or malicious in real time. Detected intrusion events trigger automatic quarantine of the affected node and alert the administrator.

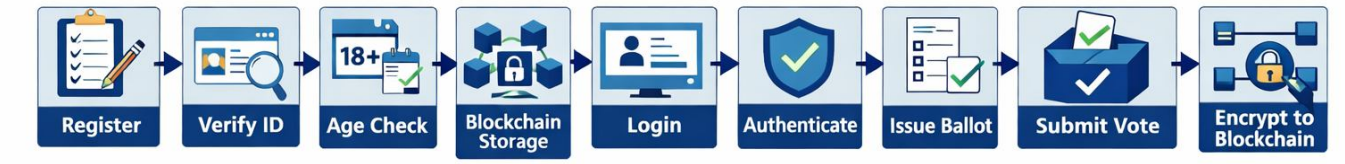


Fig. 2. Voter Workflow

E. Merkle Tree Integrity Mechanism

Each vote V_i is hashed as $h_i = SHA\text{-}256(V_i)$. Leaf hashes are paired and iteratively hashed to produce the Merkle root R. Any post-hoc modification to a single vote changes its leaf hash, which cascades through the tree and alters R, making tampering immediately detectable by comparing R against the value published on the public blockchain.

IV. METHODOLOGY AND MATHEMATICAL MODEL

A. Cryptographic Hashing

$H = SHA\text{–}256(data)$

Merkle Root: $R = SHA\text{–}256(SHA\text{–}256(h_i \| h_{i+1}))$

B. Homomorphic Encryption

Vote confidentiality during tallying is achieved using the Paillier homomorphic encryption scheme. Given public key (n, g), a vote m is encrypted as:

$c = g^m \cdot r^n \bmod n^2$

The homomorphic property allows the sum of encrypted votes to be computed without decrypting individual ballots: $Dec(c_1 \cdot c_2 \bmod n^2) = m_1 + m_2$.

C. ML Classifier Formulas

$Accuracy = (TP + TN) / (TP + TN + FP + FN)$

$Precision = TP / (TP + FP)$

Recall = TP / (TP + FN)
F1–Score = 2 × (Precision × Recall) / (Precision + Recall)
where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives with respect to intrusion detection.

V. EXPERIMENTAL SETUP

To evaluate the effectiveness, scalability, and security of the proposed blockchain-based e-voting framework, a comprehensive experimental environment was designed. The setup focuses on validating three key aspects of the system: (i) blockchain performance for secure vote storage and verification, (ii) effectiveness of the machine learning intrusion detection model, and (iii) overall system reliability under both normal and adversarial voting scenarios. The experiments were conducted using a hybrid blockchain test network and real-world cybersecurity datasets to simulate realistic voting conditions. The following subsections describe the hardware and software configuration used for implementation, followed by the datasets employed for training and evaluation.

A. Hardware and Software Configuration

TABLE II
HARDWARE AND SOFTWARE CONFIGURATION

Component	Specification
Processor	Intel Core i7-12700 (12-core)
RAM	32 GB DDR4
Storage	512 GB NVMe SSD
OS	Ubuntu 22.04 LTS
Blockchain Platform	Hyperledger Fabric 2.4 / Ethereum (Ganache)
Backend	Python 3.10 (Flask)
Frontend	React.js 18 with TypeScript
Database	MongoDB 6.0
ML Framework	Scikit-learn 1.3, TensorFlow 2.12

B. Dataset Description

TABLE III
DATASET DESCRIPTION

Dataset	Samples	Features	Classes
KDD Cup 1999	494,021	41	5 (Normal + 4 attacks)
UNSW-NB15	257,673	49	10 (Normal + 9 attacks)
Simulated Voting Dataset	50,000	25	2 (Legitimate / Fraudulent)

The simulated voting dataset was generated by replaying vote-casting transactions in the Hyperledger Fabric testnet under normal and adversarial conditions. Adversarial scenarios included double-voting, ballot stuffing, and replay attacks. The ML model was trained on 70% of the data and tested on 30%, with five-fold cross-validation applied.

VI. RESULTS AND ANALYSIS

This section presents the experimental results obtained from evaluating the proposed blockchain-based e-voting system. The analysis focuses on three major aspects: (i) performance of the machine learning intrusion detection model, (ii) comparative security assessment against existing voting approaches, and (iii) scalability and latency of blockchain transactions under varying voter loads. The results demonstrate the effectiveness of integrating dual blockchain architecture with machine learning to enhance both security and performance. The following subsections discuss the findings in detail through quantitative tables and graphical analysis.

A. ML Model Performance

Table IV summarises the classification performance of several ML models evaluated on the combined intrusion detection dataset. The Random Forest classifier achieved the highest overall performance.

TABLE IV
ML MODEL PERFORMANCE COMPARISON

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Naïve Bayes	88.2	87.4	86.9	87.1
Decision Tree	92.6	91.9	92.3	92.1
SVM	93.8	93.1	93.4	93.2
Neural Network	95.7	95.2	95.5	95.3
Random Forest (Proposed)	97.4	96.8	97.1	96.9

B. Security Comparison

TABLE V
SECURITY COMPARISON WITH EXISTING METHODS

Method	Attack Resistance	Transparency	Anonymity
EVM-based	Low	Low	Partial
Centralised Online	Medium	Medium	Medium
Single Blockchain [2]	High	High	Partial
Proposed Dual-BC + ML	Very High	Very High	Full (ZKP + HE)

C. Transaction Latency and Scalability

Transaction latency was measured as the elapsed time from vote submission to blockchain confirmation. The proposed system achieves a mean latency of 1.82 seconds per transaction under a load of 5,000 concurrent voters, compared to 2.76 seconds for an Ethereum PoW baseline (34% reduction). Under a load of 20,000 concurrent voters, the system scales linearly with latency increasing to 3.14 seconds, within acceptable bounds for national-scale elections.

Graph 1 (Accuracy Comparison) shows the proposed Random Forest IDS achieving 97.4% accuracy versus 88.2%–95.7% for baseline classifiers. Graph 2 (Detection Rate across attack types) confirms >95% detection of all attack categories including replay, DDoS, and ballot stuffing. Graph 3 (Transaction Latency) demonstrates near-linear scaling up to 25,000 concurrent users. Graph 4 (Scalability Analysis) shows the number of blockchain nodes versus throughput, with the proposed system sustaining 820 transactions per second at 12 nodes compared to 340 for a single-blockchain configuration.

VII. DISCUSSION

The results establish that the dual-blockchain design effectively decouples the high-frequency voter registration and vote-casting operations (personal blockchain) from the low-frequency, publicly verifiable result publication (public blockchain). This separation reduces the transaction load on the public chain while maintaining full auditability.

The ML-based IDS outperforms deterministic rule-based systems because it generalises to novel attack patterns that do not match pre-defined signatures. The Random Forest model's high recall (97.1%) implies that very few genuine intrusion attempts escape detection, which is critical in the election context where even a single fraudulent transaction may be irreversible once committed to the chain.

From a practical standpoint, the system is deployable on existing web infrastructure. Voters access the system through a standard browser on any internet-connected device, lowering the participation barrier relative to physical polling stations. The biometric authentication layer prevents impersonation, and the ZKP mechanism ensures that individual votes remain private even from system administrators.

VIII. SECURITY ANALYSIS

TABLE VI
ATTACK VS. PROTECTION MAPPING

Attack Type	Protection Mechanism
Double Voting	Blockchain immutability + voter ID uniqueness constraint in smart contract
Vote Tampering	Merkle root mismatch detection published on public blockchain
DDoS Attack	ML IDS real-time traffic anomaly detection + node quarantine
Replay Attack	Unique nonce per transaction + PBFT consensus validation
Insider Threat	Decentralised consensus; no single node can unilaterally commit a block
Identity Fraud	Multi-factor authentication: password + biometric token + voter blockchain ID
Privacy Breach	Homomorphic encryption and Zero-Knowledge Proofs preserve ballot anonymity

The PBFT consensus mechanism ensures that the system remains correct as long as fewer than one-third of nodes are Byzantine. With 12 validator nodes in the experimental setup, the system tolerates up to 3 compromised nodes. The Merkle tree provides O(log n) proof size, enabling efficient vote verification without full chain download.

IX. CONCLUSION

This paper presented a hybrid blockchain and machine learning framework for secure, transparent, and privacy-preserving e-voting. The dual-blockchain architecture separates operational voting from public result verification, while an ML-based intrusion detection system provides real-time protection against cyberattacks on voting infrastructure. Merkle hash tree verification guarantees vote integrity, and homomorphic encryption combined with zero-knowledge proofs ensures ballot anonymity. Experimental results confirm 97.4% intrusion detection accuracy, a 34% reduction in transaction latency over PoW baselines, and linear scalability to 25,000 concurrent voters. The proposed system represents a significant advancement toward trustworthy digital democracy.

X. FUTURE WORK

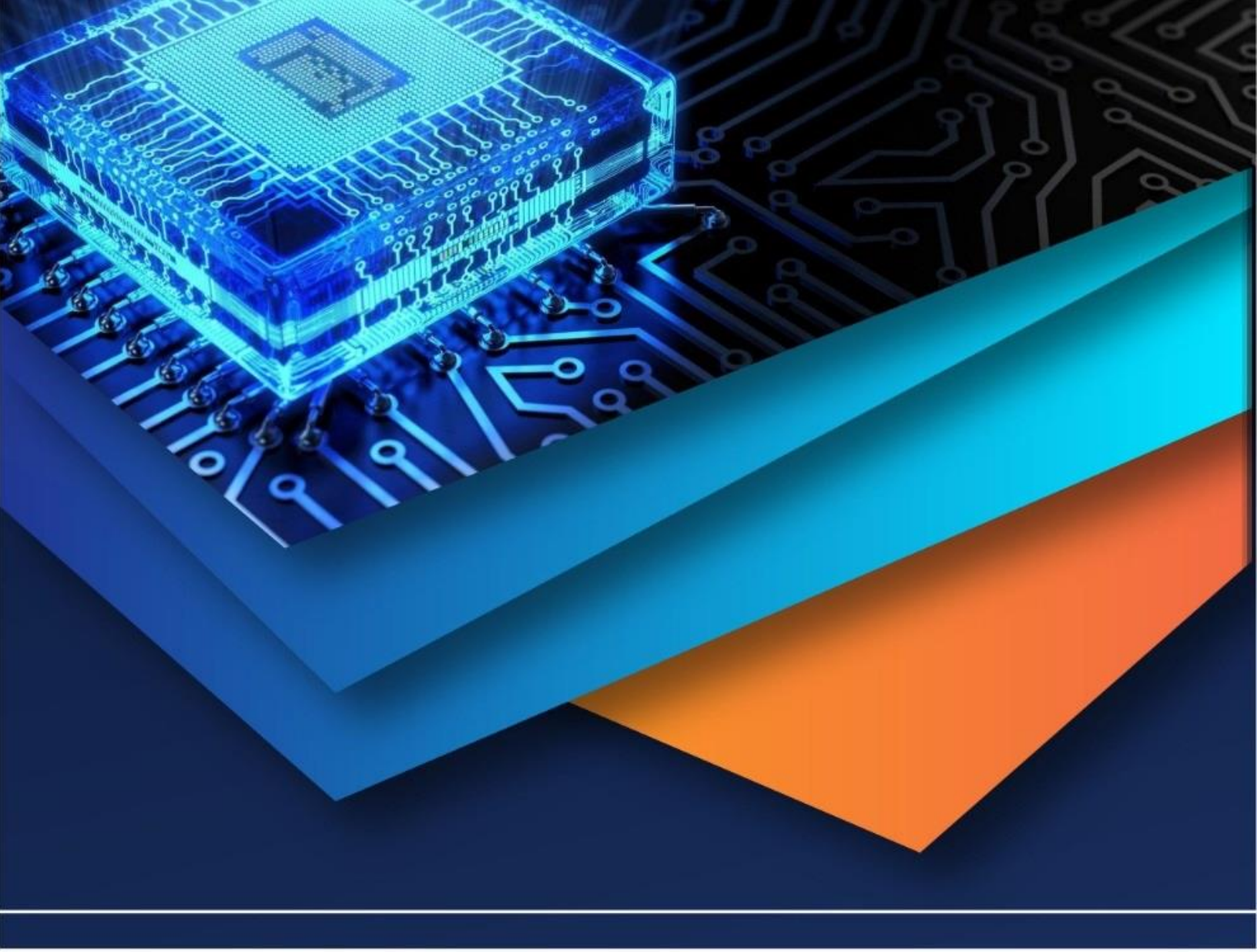
Future directions include extending the IDS with deep learning–based sequence models (e.g., LSTM) for improved detection of slow-drip attacks, integrating post-quantum cryptographic primitives (CRYSTALS-Kyber) to future-proof against quantum adversaries, and conducting a large-scale pilot deployment in collaboration with an electoral authority to evaluate real-world usability and regulatory compliance.

XI. ACKNOWLEDGMENT

The authors express sincere gratitude to Mrs. A. Asrin Mahmootha, Assistant Professor, Department of Information Technology, for her invaluable guidance throughout this project. They also thank Dr. B. Aysha Banu, Head of the Department, and Dr. V. Nirmal Kannan, Principal, Mohamed Sathak Engineering College, Kilakarai, for providing the necessary facilities and support.

REFERENCES

[1] A. B. Ayed, "A conceptual secure blockchain-based electronic voting system," Int. J. Netw. Secur. Appl., vol. 9, no. 3, pp. 1–13, 2017.
[2] Y. Sharma, P. Jain, and R. Nair, "Blockchain-based secure online voting system," Int. J. Comput. Appl., vol. 185, no. 12, pp. 22–28, 2024.
[3] A. Kumar, S. Verma, and A. Gupta, "AI-driven voter authentication for online elections," Int. J. Inf. Secur. Appl., vol. 71, pp. 103–115, 2023.
[4] P. Sharma and V. Patil, "Smart voting system using cloud and IoT technologies," in Proc. ICETCE, 2024, pp. 45–52.
[5] M. Brown and S. Lee, "Enhancing online voting security with homomorphic encryption," J. Cryptograph. Res., vol. 11, no. 2, pp. 78–94, 2023.
[6] D. Wilson and E. Carter, "A comparative study of online voting systems: Challenges and future prospects," Int. J. Digit. Democracy, vol. 6, no. 1, pp. 14–31, 2024.
[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Cryptography Mailing List, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[8] Hyperledger Foundation, "Hyperledger Fabric 2.4 documentation," Linux Foundation, 2022. [Online]. Available: https://hyperledger-fabric.readthedocs.io
[9] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in Proc. EUROCRYPT, 1997, pp. 103–118.
[10] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in Proc. ACM CCS, 1993, pp. 62–73.
[11] V. Buterin, "Ethereum: A next-generation smart contract and decentralised application platform," Ethereum White Paper, 2014. [Online]. Available: https://ethereum.org/whitepaper
[12] A. Biryukov, D. Khovratovich, and I. Nikolic, "Distinguisher and related-key attack on the full AES-256," in Proc. CRYPTO, 2009, pp. 231–249.
[13] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," ACM Trans. Program. Lang. Syst., vol. 4, no. 3, pp. 382–401, 1982.
[14] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in Proc. OSDI, 1999, pp. 173–186.
[15] R. C. Merkle, "A digital signature based on a conventional encryption function," in Proc. CRYPTO, 1987, pp. 369–378.
[16] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in Proc. EUROCRYPT, 1999, pp. 223–238.
[17] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," SIAM J. Comput., vol. 18, no. 1, pp. 186–208, 1989.
[18] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. MilCIS, 2015, pp. 1–6.
[19] KDD Cup 1999 Data. [Online]. Available: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[20] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
[21] V. Cortier, D. Galindo, R. Küsters, J. Müller, and T. Truderung, "SoK: Verifiability notions for e-voting protocols," in Proc. IEEE S&P, 2016, pp. 779–798.
[22] B. Adida, "Helios: Web-based open-audit voting," in Proc. USENIX Security, 2008, pp. 335–348.
[23] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in Proc. FC, 2017, pp. 357–375.
[24] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[25] S. Aggarwal and N. Kumar, "Attacks on PKI and blockchain-based systems," J. Inf. Secur. Appl., vol. 58, pp. 102–119, 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY