iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67653

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 13 Issue III Mar 2025- Available at www.ijraset.com

Decentralized and Biometric-Authenticated E-Voting System: A Blockchain-Based Approach

KodeLakshmi Durga Sindhujasri[1], Kaduputla Manogna[2], Sutrayeth Hari Yuktha Nanda[3], Baligiri Thandava Krishna[4], Attru Hanumantharao[5]
Aditya College Of Engineering and Technology, India

Abstract: ***Elections play a fundamental role in any democratic system, and ensuring their integrity is of utmost importance. Traditional voting methods, such as paper ballots and Electronic Voting Machines (EVMs), suffer from various limitations, including security vulnerabilities, vote tampering, low voter turnout, delays in result processing, and a lack of transparency. Digital voting solutions offer convenience but raise concerns regarding data security and susceptibility to cyber threats. Blockchain technology presents a promising solution to these challenges by providing a decentralized, transparent, and tamper-proof framework for conducting elections. As a distributed ledger system, blockchain records transactions in an immutable and verifiable manner, ensuring the integrity of votes. Key features such as decentralization, cryptographic security, transparency, and anonymity make blockchain a robust choice for secure e-voting. In this paper, we propose and implement a blockchain-based e-voting system using Ethereum smart contracts and Web3.js. Our system enforces single-use voting credentials, preventing duplicate votes, and leverages gas fees to mitigate fraudulent voting attempts. Additionally, we develop a web-based application that demonstrates the practical implementation of blockchain voting, discussing its advantages, challenges, and limitations in real-world scenarios.***

Keywords: E-voting, Blockchain, Smart Contracts, Ethereum, Decentralized Voting, Secure Elections

I. INTRODUCTION

Blockchain technology has gained significant attention with its integration into various domains beyond cryptocurrencies. Initially introduced through Bitcoin, the first digital currency, blockchain has evolved into a revolutionary technology due to its decentralized and transparent nature. This distributed ledger system ensures that data remains immutable and verifiable across multiple nodes, making it a promising solution for secure applications. In the context of e-voting, blockchain eliminates the need for a central authority, ensuring integrity and trust in the electoral process. By leveraging its peer-to-peer network, transactions and records are securely maintained, reducing the risks of tampering and fraudulent activities.

Beyond financial transactions, blockchain technology enables secure storage and management of various types of structureddata, making it applicable to domains such as identity verification, land records, healthcare, and voting systems. By utilizing cryptographic techniques, data integrity and security are ensured, preventing unauthorized modifications. Ethereum, a blockchain platform introduced after Bitcoin, extends the capabilities of blockchain by allowing the deployment of smart contracts—self-executing programs stored on the blockchain. These smart contracts ensure automation, immutability, and transparency, making them highly suitable for applications like e-voting. Once deployed, they operate independently without external control, ensuring a secure and tamper-proof voting process.In the context of e-voting, each vote is recorded as a transaction within a block. As more votes are cast, the block accumulates transactions until it reaches capacity. Once full, the block is securely linked to the existing blockchain in a sequential and immutable manner. This ensures that all voting records are permanently stored in chronological order, preventing tampering and enhancing transparency.

The initial block in a blockchain is known as the ‘Genesis block’ or ‘Block 0’. The genesis block is typically hardcoded into the system and does not reference any previous block. Once the genesis block is initialized, ‘Block 1’ is created and linked to it. Each block contains transaction data, where individual transactions (TX1, TX2, TX3, TX4) are first hashed. These transaction hashes are then combined in pairs to generate intermediate hashes (Hash0, Hash1, Hash2, Hash3). The process continues, merging the hashes further (Hash01) until a single hash, known as the Merkle root (Hash0123), is formed. This Merkle root is stored in the block header and ensures data integrity within the blockchain (Figure 1).

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

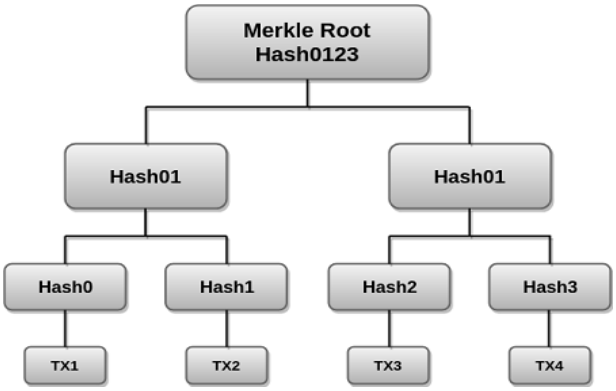


Figure1-Hashtable

The block header is where the Merkle root is stored, along with the previous block hash and a nonce value. Each block maintains a reference to the hash of the previous block, ensuring the integrity of the blockchain. Transactions within a block are hashed and structured in a Merkle tree, where individual transaction hashes (TX1, TX2, TX3) are combined to form higher-level hashes (Hash of TX1, Hash of TX2, Hash of TX3) until a single Merkle root is generated. A blockchain operates in a peer-to-peer network where nodes communicate and exchange transaction data. When a new node joins, it must perform an initial block download, synchronizing with the blockchain by validating all blocks from Block 1 to the latest block. This decentralized structure ensures that transaction data remains immutable and verifiable across the network (Figure 2).

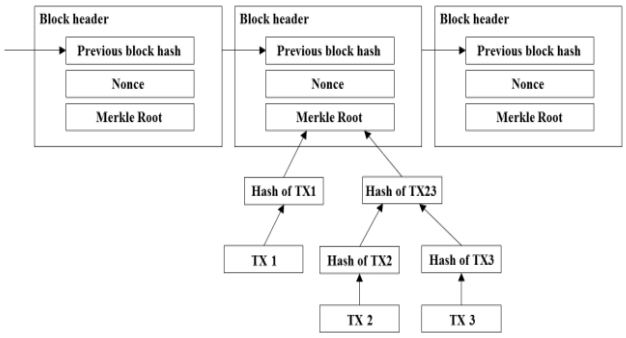


Figure-2Simplifiedbitcoinblockchain

Blockchain technology presents a promising solution for e-voting systems, offering enhanced security, transparency, and immutability. While e-voting has been extensively researched, only a few implementations have proven reliable enough for long-term use. Although online polls and surveys have been successfully deployed, implementing secure online elections for governments and businesses remains a challenge. Official elections play a crucial role in democratic governance, where ensuring transparency and voter privacy is paramount. A trustworthy electoral system is essential to maintaining public confidence in democratic processes. In the modern world, an increasing number of decisions are made through collective voting, emphasizing the need for a secure and tamper-proof voting mechanism.

The existing voting system brings up several concerns regarding its reliability and transparency. Questions arise about whether votes remain unaltered before being counted and how the integrity of the process can be verified. To address these issues, this paper explores and presents a web-based application utilizing blockchain technology on the Ethereum network, incorporating smart contracts for security and trust. Section 2 provides an overview of the current e-voting system implementations, followed by an analysis of their limitations and challenges, highlighting the need for a more robust and tamper-proof voting mechanism.

II. MOTIVATION AND RELATED WORKS

The primary objective of this project is to establish a secure voting system and demonstrate that a trustworthy e-voting mechanism can be achieved through blockchain technology. With e-voting accessible to anyone with a computer or mobile device, administrative decisions can be directly influenced by the public, or at the very least, their opinions will become more visible to policymakers and officials. This could ultimately pave the way for a more direct democratic process. Elections are often vulnerable to corruption, particularly in smaller towns and even in larger cities within unstable regions. Moreover, traditional large-scale elections incur significant costs over time, especially with the need to manage numerous polling stations and accommodate millions of voters. Additionally, voter turnout tends to be low due to logistical challenges, such as individuals not residing at their registered address or being away for personal reasons. A carefully designed e-voting system has the potential to overcome these obstacles. Although the concept of e-voting predates blockchain, existing implementations have largely relied on centralized computing and storage models, which pose security and transparency concerns.

Estonia serves as a prime example, as it was one of the first countries to adopt a fully online and comprehensive e-voting system. Discussions regarding e-voting began in Estonia in 2001, and the national authorities officially launched the system in the summer of 2003. Over the years, the system has undergone numerous enhancements and modifications, making it highly secure and reliable. Estonia's e-voting mechanism relies on government-issued smart digital ID cards and personal card readers for authentication. To participate in elections, citizens can access a dedicated web portal or use a desktop application, where they can browse candidate lists and cast their votes. This system enables any eligible voter with an internet connection, a computer, and their ID card to securely vote from a remote location.

Estonian citizens can also engage in digital democracy by creating petitions and proposing new laws through the official parliamentary website (http://rahvaalgatus.ee). These petitions can be electronically signed using the government-issued smart ID card, allowing any citizen to show support. If a petition gathers enough signatures, it is formally discussed in parliament. This initiative demonstrates how technology can enhance democratic participation. However, despite its success and a nearly 30% adoption rate in recent elections, Estonia's e-voting system has certain drawbacks. Being a centralized system, it presents a single point of failure, making it susceptible to cyber threats like hacking or hijacking. For instance, Distributed Denial of Service (DDoS) attacks could disrupt the software, servers, or databases. Additionally, system administrators might exploit their position for unethical activities, such as accessing sensitive election data. Scalability is another concern; while Estonia's system functions well for its small population, its efficiency in a densely populated country like China remains uncertain. Furthermore, reliance on physical ID cards and reader devices increases production,distribution, and maintenance costs, adding logistical challenges for voters.

Switzerland is among the few nations exploring electronic voting as an alternative to traditional ballot systems. Known for its direct democracy, the country allows every citizen aged 18 and above to participate in various elections and referendums that influence key decisions. To modernize its voting process, Switzerland has been developing a remote voting system aimed at increasing accessibility and efficiency . A similar effort was seen in a pilot project conducted in another country, where blockchain technology was tested for vote tallying. Observers manually recorded votes and later stored them on a blockchain network for verification. However, while the blockchain ensured vote integrity, the overall election process remained reliant on conventional voting methods. This highlights the challenge of fully integrating blockchain into large-scale elections while maintaining security and transparency.

A similar initiative was implemented in Moscow as part of the "Active Citizen" program. In December 2017, the city introduced blockchain technology to enhance the transparency of its voting system. This program allowed citizens to participate in decision-making by voting on various community-related questions. Each proposal discussed within the community was recorded on a blockchain-based e-voting system. Once the voting process was completed, the results were securely stored in a publicly accessible ledger, ensuring auditability and preventing any unauthorized modifications. This experiment demonstrated the potential of blockchain in increasing trust and transparency in public decision-making processes.

An example of an online polling platform, rather than a fully secure e-voting system, is StrawPoll (http://www.strawpoll.me/). This free and widely used service allows users to create and participate in polls effortlessly. It highlights the convenience of digital voting, as anyone with access to the link can cast their vote. However, its security measures are minimal—authentication is weak, there is little protection against duplicate votes, and non-repudiation is not enforced. The platform relies on users' honesty rather than implementing robust security protocols. Due to these limitations, StrawPoll is unsuitable for official decision-making processes, such as electing officials or conducting legally binding votes, but it remains useful for informal surveys and casual opinion-gathering.

In this paper, we develop a system that integrates blockchain technology into the e-voting process, presenting a practical and adaptable voting protocol that operates without the need for a Trusted Third Party (TTP). Our proposed solution ensures a secure and transparent voting mechanism while addressing the key requirements of an e-voting system. By leveraging blockchain's decentralized nature, the system enhances the integrity and reliability of elections, ultimately strengthening the credibility of the voting process.

III. IMPLEMENTATION AND DISCUSSION

This section outlines the design and functionality of our **Secure** Online **E-Voting System**, which integrates **blockchain technology with biometric authentication** to ensure a secure, transparent, and tamper-proof voting process. The system allows users to **register, authenticate using fingerprint verification, and cast votes securely** while leveraging Ethereum-based smart contracts to maintain integrity. Below is an overview of the different phases involved in the application:

1) **Registration Phase**: Voters must first register on the platform by providing email and setting up a password for it. After logging in they have to provide personal details such as **Aadhaar number, full name, date of birth, gender, address, pin code, state, and country**. Additionally, biometric data, including **fingerprint** , is collected and **hashed before being stored in the cloud**. This ensures that voter authentication is unique and secure.
2) **Login & Biometric Authentication**: After registration, voters log in using their **temporary credentials (Voter ID and temporary password)**, which are issued by the admin via email. Before proceeding to vote, **fingerprint verification is required** to confirm the voter's identity. This multi-factor authentication mechanism enhances security and prevents unauthorized access.
3) **Blockchain Integration for Secure Voting**: Blockchain technology is used to ensure a **tamper-proof and decentralized voting process**. When a vote is cast, it is **recorded as a transaction on the Ethereum blockchain**. Each transaction is **hashed and linked to the previous block**, making it impossible to alter past votes. The voting data is stored in a **smart contract**, ensuring integrity and preventing manipulation.
4) **Database & Storage Management**: The system maintains a **MongoDB database** to store **voter details, election information, and candidate data**. Candidate details, including **name, party name, photo, and party symbol**, are stored in a **backend uploads folder**. However, the actual votes are **stored securely on the blockchain** rather than in the database to ensure decentralization.
5) **Ethereum & Web3.js Integration**: The **Ethereum network is utilized to implement the blockchain environment**, where votes are stored in a secure and verifiable manner. Using **Web3.js**, the frontend interacts with the Ethereum smart contracts to allow seamless voting transactions. To ensure ease of access, the transaction cost for casting votes is **covered by the admin's MetaMask account**, eliminating the need for voters to have their own MetaMask wallets.
6) **Result Processing & Transparency**: After the voting process is complete, the system automatically **tallies the votes and declares the election winner**. Since votes are stored on the blockchain, **anyone can verify the voting records** while ensuring privacy. This guarantees transparency, as voters can trust that their votes are counted accurately.

Theapplicationfollows the Model-View-Controller (MVC) architecture, a widely used design pattern that enhances modularity and maintainability. This structure ensures efficient data management, user interaction, and business logic execution, making the system scalable and secure.

- View: This is the frontend of the application where voters interact with the system. Users can register, log in, verify their identity using fingerprint and iris authentication, and cast their votes. The interface is built using Node.js and Semantic UI React, ensuring an intuitive and seamless experience.
- Controller: This layer handles the core functionality of the application. It processes user inputs, manages biometric authentication, verifies voter eligibility, and interacts with smart contracts on the Ethereum blockchain. All business logic, including voter authentication, vote casting, and blockchain transactions, is executed here.

Model: The model layer manages voter details, election data, and results. User information such as Voter ID, biometric data, and credentials is securely stored in AWS S3 and a MongoDB database. However, all vote transactions are directly recorded on the blockchain, ensuring security, transparency, and immutability.

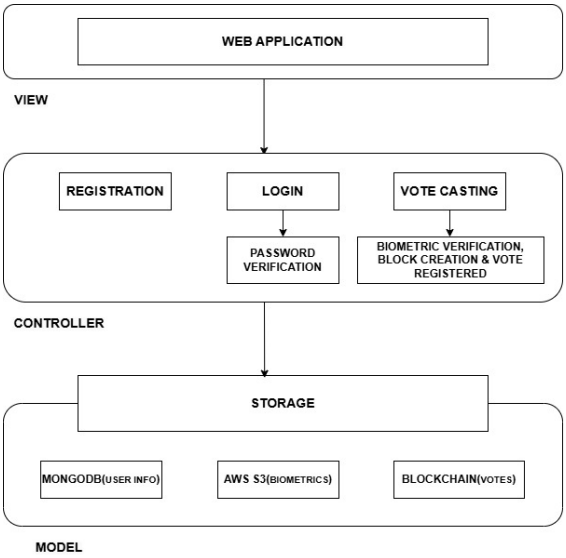


Figure-3MVCarchitecture

In general application, users are required to have a wallet address and some Ether to vote. However, since users may not be familiar with blockchain technology or managing wallets, all the vote transaction fee is on the admin's wallet. This process involves paying a small transaction fee ("gas"), which is rewarded to the miner-node that processes the transaction. While reading data such as the candidate list and election results is free, casting a vote requires a fee due to the decentralized nature of the blockchain.

To implement this, our application uses Ethereum smart contracts, which handle vote storage, validation, and election results. These contracts ensure secure voting, prevent multiple votes, and automatically count votes. The public ledger acts as the database layer, while smart contracts execute the business logic that governs voting.

The first step in developing this system is installing dependencies and deploying the smart contract. In Solidity, a contract is declared using the contract keyword, followed by defining state variables to store candidate details and votes. The constructor initializes the contract, ensuring votes are securely recorded and election results remain tamper-proof.

```
  contractElection{
    //ModelaCandidate
    struct Candidate {
            uint id;
            string
    name;
            string
    party;
string andidatePhoto;
            string
    partySymbol;
            uint
    voteCount;
        }constructor()
    public {
           addCandidate("Candidate
 1", "Party A", "photo1.jpg",
 "symbol1.png");
           addCandidate("Candidate
 2", "Party B", "photo2.jpg",
```

Fig4–Codeblocktodefinestructvariableandcontract

We have specified that struct candidate has an id of unsigned integer type, name of string type, and the vote countofunsignedintegertype.Tostorethesestructsweuse solidity mapping which is like associative array or a hash, that associates key-value pairs.

```
mapping(uint=>Candidate)publiccandidates;
```

here the key to mapping is unsigned integer and value is Candidate structure type and mapping's visibility is set to public so as to get a getter function.

The complete contractcode contains mapping, function toaddcandidatesandsmartcontractcalledcontractelection.

```
contractElection
//ModelaCandidate
struct Candidate {
        uint id;
        string name;
        string party;
        string candidatePhoto;
        string partySymbol;
        uint voteCount;
    }
// Read/write candidates mapping(uint =>
Candidate) public candidates;
// Store Candidates Count
uintpubliccandidatesCount;

function createElection(string memory
      _electionName, uint[] memory
      _candidateIds) public {

           elections[_electionName] =
     Election(_electionName,
     _candidateIds);

     emit ElectionCreated(_electionName,
     _candidateIds);

}

function addCandidate (string memory
_name, string memory _party, string memory
_candidatePhoto, string memory
_partySymbol) public {

       candidatesCount++;

       candidates[candidatesCount] =
Candidate(candidatesCount, _name, _party,
_candidatePhoto, _partySymbol, 0);

       emit
CandidateAdded(candidatesCount, _name,
_party, _candidatePhoto, _partySymbol);

}
```

Fig.5.Codeblockofcompletecontractcode

After developing the server-side application, we proceeded to build the client-side application to interact with our smart contract. The front-end was created using Reactjs. To enhance the security of the system, we implemented an additional feature, aside from the unique ID and password: temporary login credentials. These credentials are sent to users when an election is created, and they are used for voter verification. Once a voter has cast their vote, the credentials expire to prevent multiple voting.

After setting up the web application, the next step was to establish a connection with the blockchain. Unlike traditional applications where individual users connect via MetaMask, our system uses the admin's wallet to interact with the Ethereum blockchain. This approach eliminates the need for users to manage blockchain wallets or Ether.

When a user casts a vote, the transaction is signed and executed using the admin's wallet, ensuring a seamless and secure voting experience. This method simplifies the process for voters while maintaining the integrity and transparency of the blockchain network.

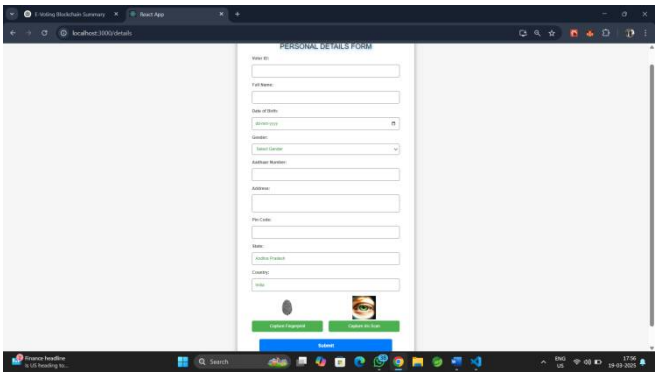


Fig6–Screenshotofapplicationduring user registrationprocess

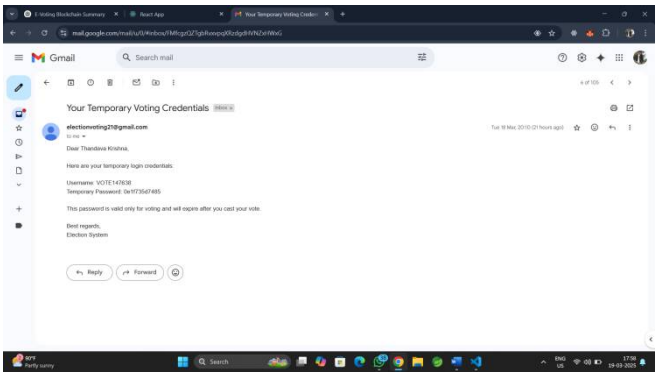


Fig7–Screenshotof user getting temporary login credentials through registered mail

The next step was to implement the functionality for users to cast their votes in the election. To track which accounts have voted, we created a mapping for voters and integrated it with the smart contract. A vote function was added, which takes the candidate-Id as an argument. The function ensures that the voter has not voted previously, verifies that the candidate is valid, and then marks the user as having voted. After the vote is cast, the candidate's vote count is updated accordingly as shown in Fig-9.

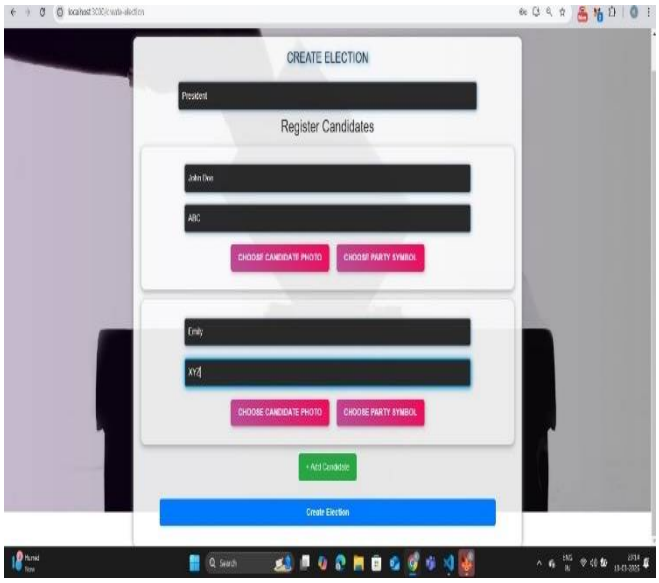


Fig8–Screenshotofapplicationduring election creationprocess


```
// Store accounts that have voted
mapping(string => bool) public hasVoted;

---

functionvote(uint _candidateId, string memory
_voterId)public{
        //requirethattheyhaven'tvotedbefore
        require(!hasVoted[_voterId], "You have already voted");
        // require a valid candidate
        require(_candidateId > 0 && _candidateId <=
        candidatesCount, "Invalid candidate");
        //recordthatvoterhasvoted
        hasVoted[_voterId] = true;
        // update candidate vote Count
        candidates[_candidateId].voteCount++;
        //triggervotedevent
        emit VoteCasted(_candidateId, _voterId);
    }
```

Fig9–CodeBockforcastingofvote/voteprocess

In our system, when a user casts their vote, the transaction is executed using the admin's wallet, which incurs a small gas fee. This fee is paid to the node responsible for validating and recording the vote on the blockchain. Once the transaction is successfully processed, the updated vote count is reflected, and the candidate with the highest number of votes is automatically declared the winner.

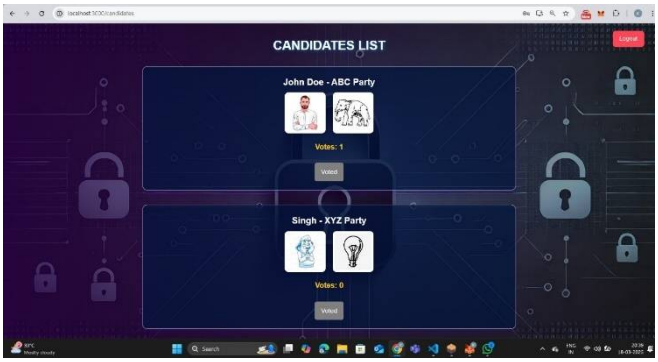


Fig.10.Screenshotofavoting page after casting vote

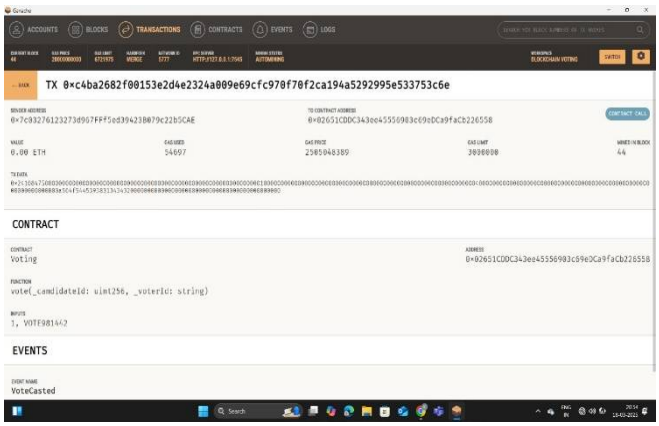


Fig.11.Screenshotofavotecastingentryinthechain.

Figure 10 shows the voting page displayed after successfully casting a vote. The interface presents the list of candidates with their respective vote counts updated immediately after the vote is recorded on the blockchain. This ensures that the voting status is accurately reflected for the user, confirming that the vote has been cast successfully.

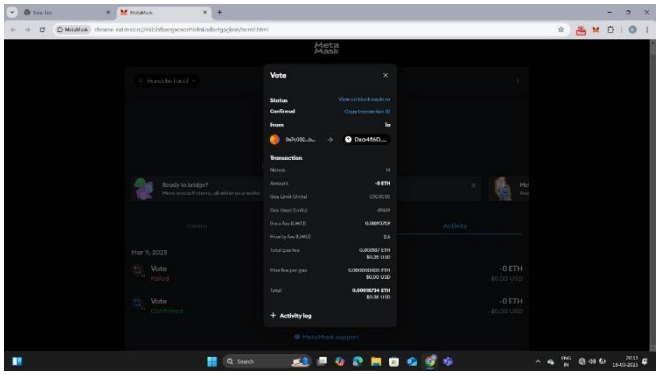


Fig.12.ScreenshotofaTransaction in MetaMask

Figure 12 illustrates the results page, which becomes available once the election is concluded and the results are announced. Registered users receive an email with a link to this page, granting them secure access to the final election results. The page displays the total vote counts and highlights the candidate with the highest votes, providing transparency and easy access for all participants. When users cast their votes, a transaction is triggered through MetaMask, connecting to the blockchain. This transaction is signed using the admin's wallet to securely store the vote on the blockchain. The admin's wallet also covers any gas fees involved, so voters do not need to manage wallets or interact with MetaMask themselves.

In this project, the scope is currently limited to small-scale elections, such as college or organizational elections. Scaling the system to handle millions of voters in large-scale elections presents unique challenges that require further research. Since the system operates on the Ethereum blockchain, the application can be accessed from any device or platform with a browser. However, ensuring voter anonymity in blockchain-based e-voting systems remains a critical challenge. All transactions, including votes, are recorded on the blockchain in plaintext, which makes it possible for anyone with access to the chain to trace votes between wallet addresses. While this transparency is a key strength of blockchain, it poses a significant disadvantage in preserving voter privacy, making such systems unsuitable for large-scale or critical elections. Addressing this issue of anonymity remains an active area of research. Several approaches, such as the use of public-private key pairs and cryptographic techniques like Diffie-Hellman, have been explored to enhance ballot privacy and enable more secure voting processes.

IV. CONCLUSION

Inthispaper,we introduced a blockchain-based electronic voting system that leverages smart contracts to provide a secure and cost-effective method for conducting elections while ensuring voter privacy. By comparing our system to traditional voting methods, we demonstrated that blockchain technology offers an innovative approach for democratic countries to transition from traditional pen-and-paper elections to a more efficient, transparent, and secure election process.

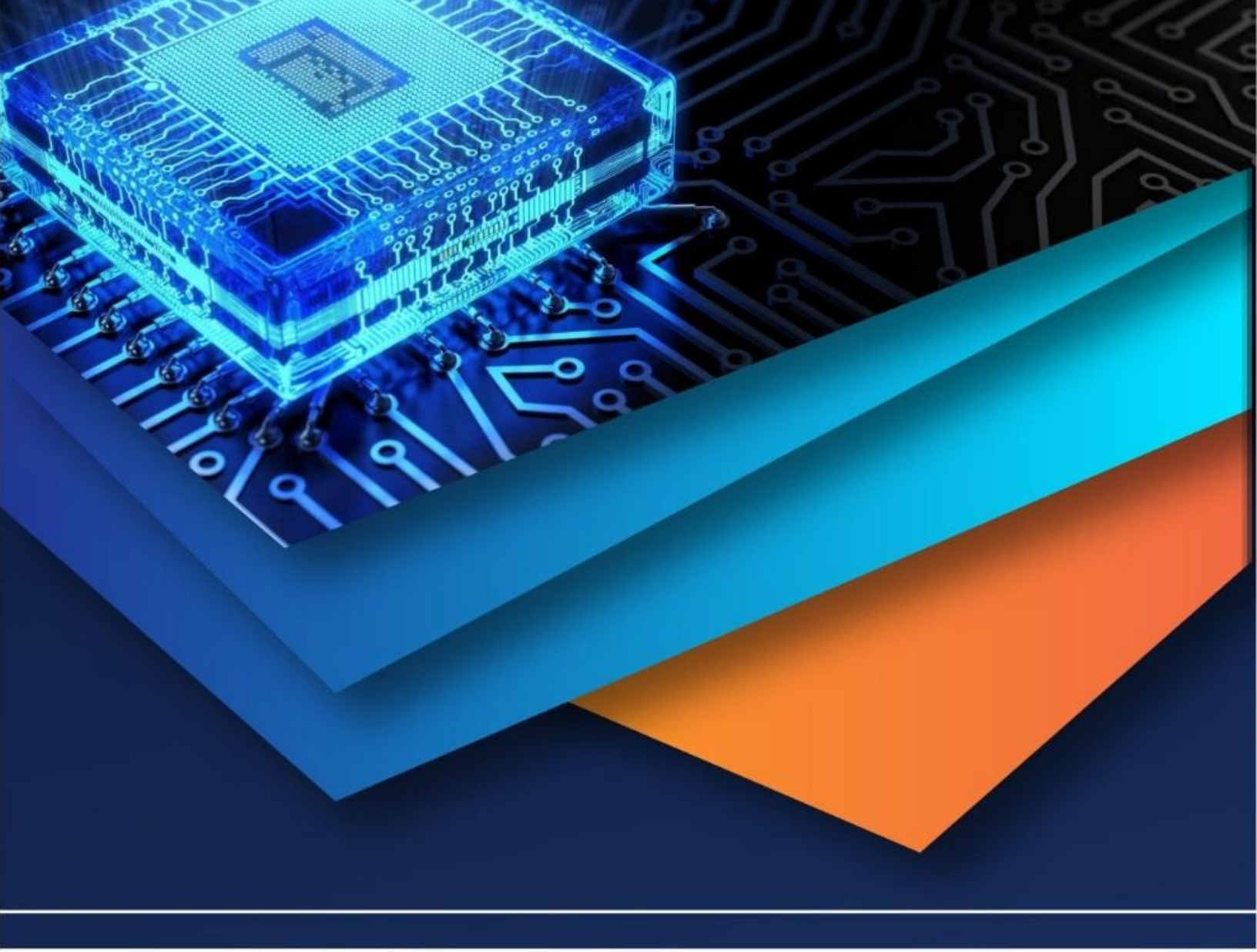
E-voting continues to be a topic of debate among both political and technical communities. While there are some successful examples, many systems have either failed to meet the security and privacy standards of traditional elections or have faced significant issues in usability and scalability. However, blockchain-based e-voting solutions, like the one we developed using Ethereum and smart contracts, address many of the critical security concerns, including voter privacy, vote integrity, verification, non-repudiation, and transparent vote counting. Despite these advances, certain aspects, such as voter authentication on a personal level, may still require additional mechanisms, such as biometric verification, to further secure the system.

Although blockchain technology holds significant promise, it is still in the early stages of development. Much more research is needed to fully realize its potential, particularly for complex applications like e-voting. Ongoing efforts to improve blockchain's scalability and security will be essential to enhance its viability for such applications.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," [Online]. Available: https://bitcoin.org/bitcoin.pdf.
[2] M. Hajian Berenjestanaki, H. R. Barzegar, N. El Ioini, and C. Pahl, "Blockchain-Based E-Voting Systems: A Technology Review," Electronics, vol. 13, no. 1, pp. 1-17, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/1/17

[3] G. Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger,” Ethereum Project Yellow Paper, vol. 151, pp. 1-32, 2014.
[4] C. D. Clack, V. A. Bakshi, and L. Braine, “Smart Contract Templates: Foundations, Design Landscape, and Research Directions,” Mar. 2017, arXiv:1608.00771.
[5] A. Tekade, R. Kantule, V. Padghan, and P. Pachpute, “E-Voting System Using Blockchain Technology,” International Journal of Engineering Research & Technology (IJERT), vol. 12, no. 5, pp. 1-10, 2023. [Online]. Available: https://www.ijert.org/research/e-voting-system-using-blockchain-technology-IJERTV12IS050175.pdf.
[6] E. Maaten, “Towards Remote E-Voting: Estonian Case,” Electronic Voting in Europe - Technology, Law, Politics and Society, vol. 47, pp. 83-100, 2004.
[7] U. C. Çabuk, A. Çavdar, and E. Demir, “E-Demokrasi: Yeni Nesil Doğrudan Demokrasi ve Türkiye’deki Uygulanabilirliği,” [Online]. Available: https://www.researchgate.net/publication/308796230_E-Democracy_The_Next_Generation_Direct_Democracy_and_Applicability_in_Turkey.
[8] F. Hao and P. Y. A. Ryan, Real-World Electronic Voting: Design, Analysis and Deployment, CRC Press, pp. 143-170, 2017.
[9] N. Braun, S. F. Chancellery, and B. West, “E-Voting: Switzerland's Projects and Their Legal Framework – In a European Context,” Electronic Voting in Europe: Technology, Law, Politics and Society, Gesellschaft für Informatik, Bonn, pp. 43-52, 2004.
[10] P. McCorry, S. F. Shahandashti, and F. Hao, “A Smart Contract for Boardroom Voting with Maximum Voter Privacy,” International Conference on Financial Cryptography and Data Security, Springer, Cham, pp. 357-375, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY