IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025

DOI: 10.17148/IJARCCE.2025.14347

Secure and reliable E-Voting using Blockchain technology

G. Shireesha M Tech[1], M V S S Nanda Kishore[2], K. Venkat Reddy[3],K Sandeep Kumar[4]

Information Technology, Vasireddy Venkatadri Institute of technology,Guntur,India[1-4]

Abstract: In every nation, democratic voting is an important and serious event. Currently, ballot papers or electronic voting machines are used for voting. These procedures have several disadvantages, including a lack of transparency, low voter turnout, vote tampering, mistrust of the electoral authority, voter ID card forgeries, result delays, and security concerns. Security is always the top priority when considering putting a digital voting system in place. There can be no question regarding the system's capacity to protect data and fend off possible attacks when such important choices are on the line. Blockchain technology is one possible solution to the security problems. There are countless uses for blockchain technology. Blockchain is a distributed ledger technology that facilitates peer-to-peer, decentralized network transactions involving digital assets. One interesting development in this area is distributed ledger technology. A block is a grouping of every transaction. Immutability, decentralization, security, transparency, and anonymity are some of the key characteristics of blockchain technology. A promising option for creating more transparent, safe, and secure electronic voting systems is blockchain technology with smart contracts. In this paper, we have used the Solidity language and blockchain technology to implement and test a sample e-voting application as a smart contract for the Ethereum network using wallets. To prevent vote duplication, a limited quantity of tokens (gas) are provided in the wallet and depleted when the user casts their ballot. In addition to outlining the benefits and drawbacks of utilizing blockchain technology, this paper presents a workable system by highlighting a voting web app and its restrictions.

Keywords: E-voting, Smart-contracts, Blockchain, Ethereum.

I. INTRODUCTION

Voting is a fundamental element of any democratic society. The integrity of the electoral process directly impacts public trust in governance. Traditional voting systems, including paper ballots and EVMs, suffer from various limitations:

- Lack of transparency and traceability
- Vulnerability to manipulation and fraud
- Low voter turnout due to logistical issues
- Delays in result processing

Blockchain technology provides an ideal solution by ensuring decentralization, immutability, and security. A blockchain-based e-voting system can record each vote as a transaction, securing it in a public ledger that cannot be altered. This guarantees vote integrity and real-time tracking. Smart contracts on the Ethereum blockchain automate the voting process, removing the need for intermediaries and increasing trust among stakeholders.This paper outlines a secure and transparent e-voting framework using blockchain technology. The proposed system ensures end-to-end transparency, real-time vote verification, and enhanced voter security. Each block also retains a record of the header of the previous block to guarantee that a transaction cannot be changed; therefore, to alter data, you would have to. Blockchain has important features like immutability, decentralization, security, transparency, and anonymity. When combined with smart contracts, blockchain technology presents itself as a promising candidate for creating safer, more transparent, and secure electronic voting systems. In this paper, we have used the Solidity language and blockchain technology to implement and test a sample e-voting application as a smart contract for the Ethereum network using wallets. Only a small number of implementations, nevertheless, are still in use and sufficiently dependable. Of course, there are plenty of successful examples of online surveys and polls, but we can't say the same about online elections for corporations and governments.People (and members of organizations) make a lot of decisions these days Many questions are raised by the current voting system, including how reliable and transparent the system is, whether the votes are not changed before they are counted, and how we can verify the transparency of the system. Therefore, we explore and suggest a web application that uses blockchain technology over the Ethereum server by deploying smart contracts to address this type of question in this paper. The current implementation of electronic voting systems and its limitations are covered in the second section.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025

DOI: 10.17148/IJARCCE.2025.14347

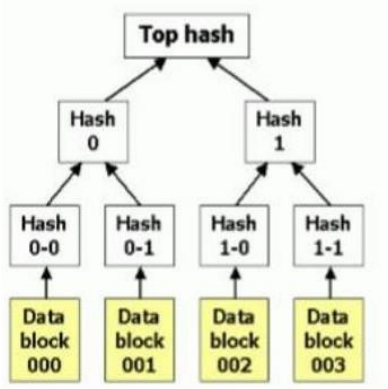


Fig 1- Hash table

The block header is where the merkle root is stored. To ensure that a transaction cannot be modified each block also keeps a record of the previous blocks header, this means to change data you would have to A blockchain is designed to be accessed across a peer-to-peer network, each node/peer then communicates with other nodes for block and transaction exchange. Once connected to the network, peers start sending messages about other peers on the network, this creates a decentralised method of peer discovery. The purpose of the nodes within the network is to validate unconfirmed transactions and recently mined blocks, before a new node can start to do this it first has to carry out an initial block download. The initial block download makes the new node download and validate all blocks from block 1 to the most current blockchain, once this is done the node is considered synchronised

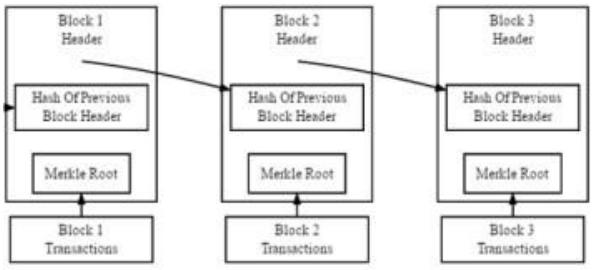


Figure-2 Simplified bitcoin blockchain(source-bitcoin.org)

Blockchain might be a suitable solution for e-voting projects. E-voting is being studied extensively, and many implementations are tested and even used for a while. However, very few implementations are reliable enough and are still in use. Of course, there are many successful examples of online polls and questionnaires, yet we cannot claim the same for online elections for governments and businesses. That's mainly because, official elections are essential elements of the democracy and democratic administrations, which are the most preferred administrative methodology in the modern world. More, what is most valued in democratic societies is a robust electoral process that provides transparency and privacy. Today, a lot of decisions are being made by people (and members in organizations)[2]. The current voting scheme raises many questions such as how reliable and transparent the system is, are the votes not changed before they are counted, how can we verify the transparency of the system. So to tackle this kind of questions in this paper we investigate and propose a web application using the blockchain technology over the Ethereum server by deploying smart contracts. In the 2 section we discuss about the current deployment of e-voting systems and further on we discuss limitations of it.

II. MOTIVATION AND RELATED WORKS

Our primary goal in this project is to demonstrate that a trustworthy electronic voting system can be implemented using blockchain technology while also offering a secure voting environment. Because every administrative decision can be made by individuals and members when electronic voting is accessible to everyone with a computer or a mobile phone, or at the very least, people's opinions will be more widely known and easier for managers and politicians to access. Humanity will eventually arrive at true direct democracy as a result of this [6]. It's crucial for us because elections are susceptible to manipulation and corruption, particularly in small towns and even larger cities in corrupt nations. Furthermore, the long-term costs of large-scale traditional elections are high, particularly when millions of voters are involved and hundreds of vote centers are spread out geographically [7]. Additionally, there is a low voter turnout at the

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025
DOI: 10.17148/IJARCCE.2025.14347

polling places because the voter may not be residing at the address where his name is on the list, or he may be on vacation or at another job. With careful implementation, e-voting can resolve these issues. Blockchain is much more recent than the idea of electronic voting.

Therefore, centralized computation and storage models have been used in all known examples to date. Since the Estonian government was among the first to introduce a fully online and comprehensive e-voting solution, it is an excellent example [8]. In 2001, the idea of electronic voting began to be discussed in the nation, and in the summer of 2003, national authorities formally launched it [9]. Despite numerous changes and enhancements to the original plan, their system is still in use today. As stated, it is presently very dependable and robust. For person-wise authentication, they make use of government-distributed smart digital ID cards and personal card readers [10].

Both an equivalent desktop app and a dedicated web portal are available for citizens to use to list the candidates and cast their votes during the elections. In the field of employment, individuals can also electronically draft petitions and proposals for legislation on the parliament's website (http://rahvaalgatus.ee). Any citizen who wishes to support the proposal can use the smart ID card to digitally sign these petitions. Proposals are discussed in parliament if a specific number of signatures is obtained. That is yet another excellent illustration of how technology can support democracy. The Estonian model has certain disadvantages despite its significant success and recent elections, when it achieved a penetration rate of almost 30%. By its very nature, the centralized solution introduces a single point of failure and leaves room for hacking and hijacking attempts. Distributed Denial of Service (DDoS) attacks, for instance, have the potential to damage servers, databases, and software.

During an election, the administrators of such a system might act malevolently and steal, if they are unable to manipulate, some important information. Another concern is this system's scalability.It is difficult to predict whether such a system would function perfectly in, say, China because of Estonia's small population. Due to the additional expense of creating, distributing, and carrying (for voters) the ID card and reader device, the ongoing requirement is also not pleasant. One of the few nations following the trend of electronic voting is Switzerland. Every citizen of Switzerland, a country renowned for its extensive democracy, is eligible to participate actively or passively in elections that may be held on a wide range of issues and for a wide range of decisions. Additionally, they have formally started working on a voting system known as remote voting [11].

For instance, the general election in March 2018 in Sierra Leone Agora, a Swiss startup, conducted counting in two districts. Following the vote, roughly 400,000 ballots were manually entered into Agora's blockchain system by a group of certified observers from various locations. Similar commercial or experimental work was carried out in the Russian city of Moscow for its Active Citizen program in December 2017.

These systems were a partial implementation of blockchain, and the votes were verified by blockchain. To make the voting results publicly auditable, the program began utilizing a blockchain for voting. A blockchain is used to transfer each community-discussed question that is up for vote to the electronic voting system. The results are displayed on a ledger that includes all of the prior polls once the voting is finished [10]. Instead of using an electronic voting system, http://www.strawpoll.me/ is a well-known and cost-free online polling service. It's a straightforward website where anyone can make surveys and vote to respond to others' polls. Because everyone can easily access the election, cast their vote, and declare their choice, it demonstrates the power of electronic voting.

As long as they know the link, people can share private hyperlinks to any poll that has been created. Only those who have the link can cast a vote, and a single browser can only cast one vote. There is very little security in this place about voter authentication, duplicate votes, and vote non-repudiation. http://www.strawpoll.me/ gives people confidence that they won't tamper with the election process while taking advantage of the convenience and capabilities of electronic voting. As a result, it cannot be applied in practical situations like selecting a department chairman, etc. [2]. In this paper, we develop a system by incorporating the blockchain concept into the electronic voting process and develop a workable and universal electronic voting protocol that does not require a TTP. This system offers a safe and adaptable voting mechanism that meets nearly all of the primary requirements for an electronic voting system and enhances electoral power.

III. IMPLEMENTATION AND DISCUSSION

In this section we will illustrate the design and functional phase of our application, The User accesses the web application where the platform is hosted and register's itself as well as cast its vote in an secured and transparent manner. Fig 3 depicts the overview of the application.

Registration Phase: First, the voter must register using their unique ID and personal information, including their name, roll number, and mobile number. The database contains all of this information.

1. Login: To cast a ballot, the voter attempts to log in after registering. Voters first use a password to log in during this phase. Voters must authenticate themselves after successfully logging in before they can cast their ballot. OTP verification is used for real-time authentication to increase security.
2. Blockchain Technology: The primary function of this technology is security. Blockchain offers a transparent

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025

DOI: 10.17148/IJARCCE.2025.14347

and safe environment. The voter message (cast vote) is encrypted by the blockchain using an asymmetric encryption algorithm. Blockchain provides the public key, while the host has the private key. The ledger uses the public key for verification.

3. Database: The database contains the user database. Names, genders, and unique IDs are among the details that are kept in a database. The suggested database to be used is MySQL.

4. Ethereum Network: The Ethereum network offers a structure for building and storing blockchains. An encrypted ledger is used to create each block and store its details. The system has high fault tolerance, thanks to the distribution of these generated blocks among nodes.

5. Results phase: This stage involves processing and tallying the votes. The website displays the results that have been generated. Voters can use their public key to validate their votes. This makes the voting process more transparent.

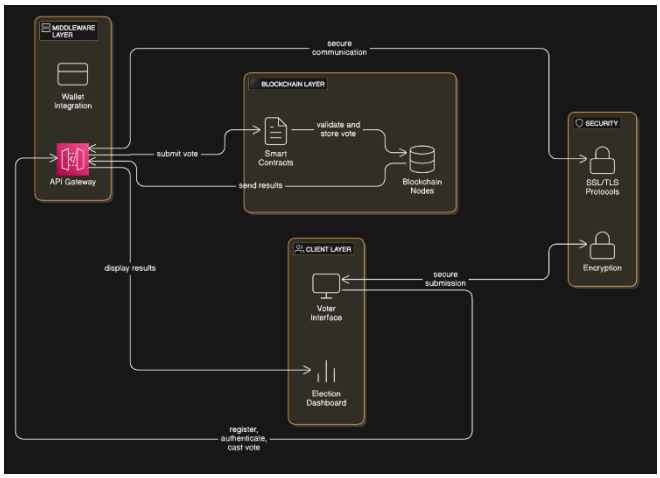


Fig 3 – System Overview

The application is built using the architectural pattern of Model-View-Controller. It is also widely used architecture. Here, the application is divided into three main logical components: the model, the view and the controller.

• View: The top layer is where the end-user communicates with the application through clicking buttons, typing details, accessing camera, selecting radio button, uploading songs, etc. This layer is responsible for displaying all data or a portion of data to user based on the requirement of the application. This layer also acts as a bridge between the user and application itself.

• Controller: This middle layer of the application contains the business logic, and the main functionality of the application. As soon as the user interacts with the application,the response is processed in this layer. From log-in to casting vote, all the functions that run in background belong to this layer. This mainly consists of all the functions and sending output to view layer

• Model: This layer is responsible for maintaining the user's data. Relational Database MySQL is used for storing user data. deploy the contract. shows the structure, variable and contract declaration.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025

DOI: 10.17148/IJARCCE.2025.14347

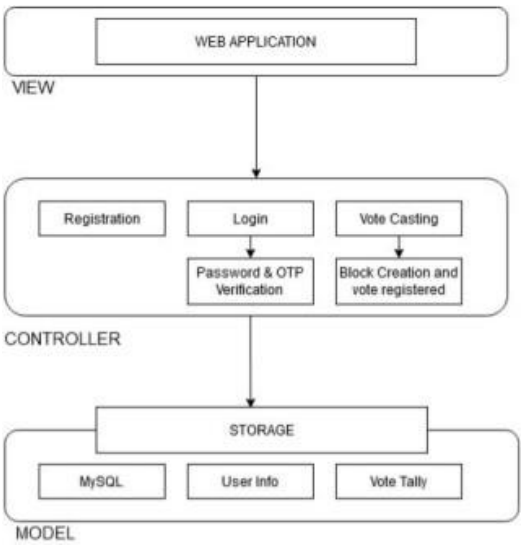


Fig-4 MVC architecture

In our application for a user to vote he/she needs an account with a wallet address and some Ether, Ethereum's cryptocurrency. After connecting to the network they cast their vote and pay a small transaction fee to write their vote to the blockchain. This transaction fee is called as "gas" in our application which can be related to some coins. This transaction fee "gas" is awarded to the miner-node of the network after he completes the transaction. It's important to note that voting on the blockchain costs us some Ether but seeing the list of candidates is free, because writing to blockchain costs but reading data from the blockchain is free.

To code our application Ethereum blockchain allows us to execute code with Ethereum Virtual machine (EVM) on blockchain with smart contract. In our application Smart contracts are responsible of reading and writing data to the blockchain as well as executing the logic. Smart contracts are written in programming language called Solidity. If the public ledger represents database layer of the blockchain, then smart contracts are where all the business logic that transacts with that data lives. Smart contracts represent a covenant or agreement, In our application its is an agreement that user's vote will count, others vote will be counted only once and the candidates with highest vote will be declared the winner.

Step first to build our application is installing all the dependencies and then writing our contract and deploying it to the blockchain successfully. To create the contract declare the smart contract with the "contract" keyword, followed by the contract name. Next, we declare a state variable that will store the value of the candidate name. State variables allow us to write data to the blockchain constructor is called whenever we

```
contract Voting {
    You, 2 weeks ago | 1 author (You)
    struct Election {
        uint256 startTime;
        uint256 endTime;
        bool isActive;
        bool isPublished;
        mapping(uint256 => uint256) candidateVotes; // candidateId => votes
    }
```

Fig 5 – Code block to define struct variable and contract

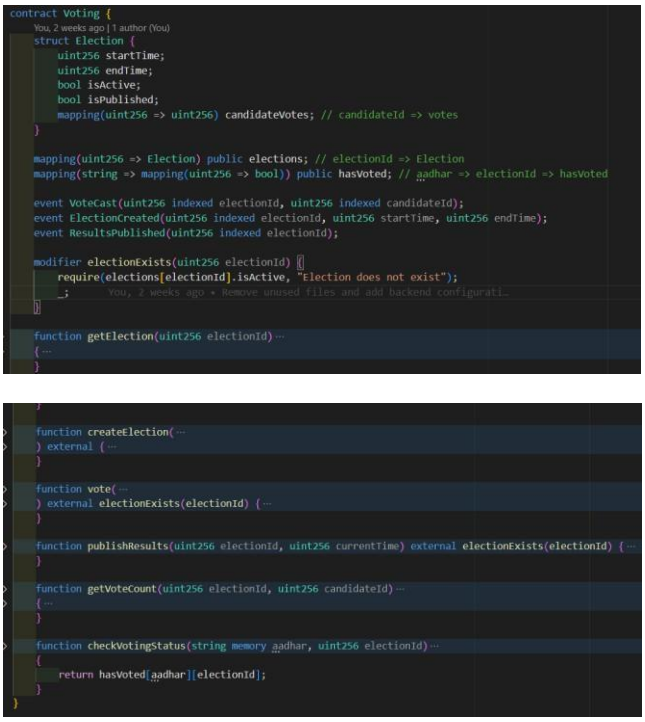
We have specified that struct candidate has an id of unsigned integer type, name of string type, and the vote count of unsigned integer type. To store these structs we use solidity mapping which is like associative array or a hash, that associates key-value pairs. here the key to mapping is unsigned integer and value is Candidate structure type and mapping's visibility is set to public so as to get a getter function. The complete contract code contains mapping, function to add candidates and smart contract called contract election

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025

DOI: 10.17148/IJARCCE.2025.14347



```
contract Voting {
    You, 2 weeks ago | 1 author (You)
    struct Election {
        uint256 startTime;
        uint256 endTime;
        bool isActive;
        bool isPublished;
        mapping(uint256 => uint256) candidateVotes; // candidateId => votes
    }

    mapping(uint256 => Election) public elections; // electionId => Election
    mapping(string => mapping(uint256 => bool)) public hasVoted; // aadhar => electionId => hasVoted

    event VoteCast(uint256 indexed electionId, uint256 indexed candidateId);
    event ElectionCreated(uint256 indexed electionId, uint256 startTime, uint256 endTime);
    event ResultsPublished(uint256 indexed electionId);

    modifier electionExists(uint256 electionId) {
        require(elections[electionId].isActive, "Election does not exist");
        _;      You, 2 weeks ago • Remove unused files and add backend configurati…
    }

    function getElection(uint256 electionId)···
    {···
    }

    function createElection(···
    ) external {···
    }

    function vote(···
    ) external electionExists(electionId) {···
    }

    function publishResults(uint256 electionId, uint256 currentTime) external electionExists(electionId) {···
    }

    function getVoteCount(uint256 electionId, uint256 candidateId)···
    {···
    }

    function checkVotingStatus(string memory aadhar, uint256 electionId)···
    {
        return hasVoted[aadhar][electionId];
    }
}
```

Fig. 6. Code block of complete contract code

After creating the server side application we created client side application that will talk to our smart contract. we created our front-end with java script and HTML. To make our system more secure we have included one more unique feature other than unique id and password is the OTP(one time password) feature. We request users to enter their mobile no on which otp is send and then the system verifies the user. After creating the webpage we need to log in to the blockchain. To connect to blockchain we need to import one of the accounts from ganache- One of the dependencies which gives us 10 accounts with account address and some fake ethers, into MetaMask.

In order to use blockchain we must install a special browser extension in order to use the Ethereum blockchain. That's where MetaMask is used. After connecting we can interact with our smart contract and will be able to see our contract and account data

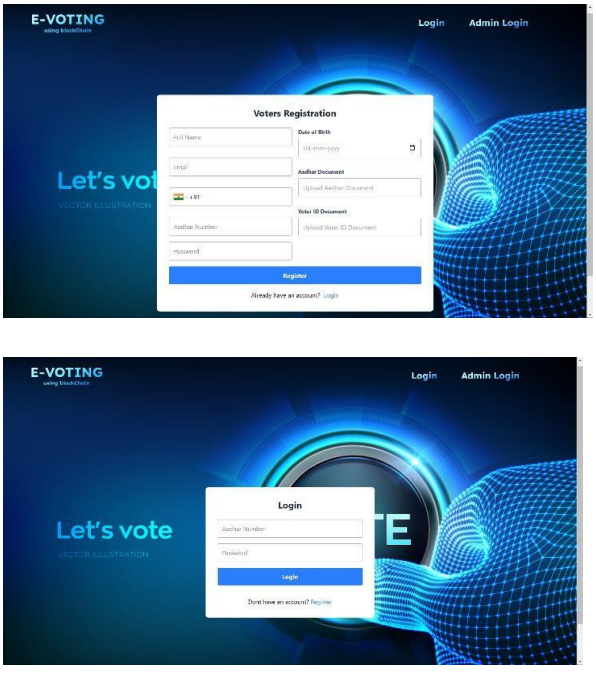


Fig 8 – Screenshot of application during registration process

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025

DOI: 10.17148/IJARCCE.2025.14347

The next step was to add the ability to cast votes in the elections. To keep the track of accounts that have voted we define voters and mapped it to the smart contract, and add 'vote' function which takes in one argument- candidate-Id. It checks that the user hasn't voted before, candidate is valid, recording that user has voted after his voting and then update the candidate vote count. Fig-9 depicts the code and mapping for casting the vote

```
function vote(
    uint256 electionId,
    uint256 candidateId,
    string memory aadhar,
    uint256 currentTime
) external electionExists(electionId) {
    require(
        currentTime >= elections[electionId].startTime,
        "Election has not started"
    );
    require(
        currentTime <= elections[electionId].endTime,
        "Election has ended"
    );
    require(
        !hasVoted[aadhar][electionId],
        "Already voted in this election"
    );

    elections[electionId].candidateVotes[candidateId]++ ;
    hasVoted[aadhar][electionId] = true;

    emit VoteCast(electionId, candidateId);
}
```

Fig 9 – Code Bock for casting of vote & vote process when the user votes by using gas which is rewarded to the node(miner) whoever writes it to the blockchain, after successful casting of votes results are displayed and candidate with highest votes is the winner.

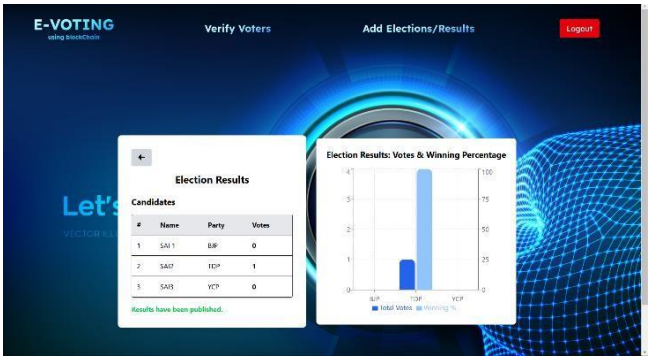


Fig. 10. Screenshot of a Election results

Fig 10 shows the elections results after successful casting of vote and fig 11 shows vote casting entry in the chain with transaction hash, blocks-created till now, contract address, timestamp, account, block number of the transaction, gas used, and the total cost during the whole process of casting. In this project, our scope is limited for small-scale polls and elections such as college elections. A larger voting with millions of voters may have different problems to address. The Ethereum network's scalability is still unknown and needs further research, that's why we cannot suggest use of these contracts for nation-wide elections, at least for now.

Our contracts are executed in the Ethereum blockchain, so wherever browser can be run (location, platform, device, etc.), our voting application can be used, too. A fundamental problem of blockchain based e-voting systems is to provide anonymity for voters without compromising the transparency of the general voting process. In detail, all the transactions (money transfers, votes etc.) are essentially written to the blocks of the blockchain as plaintext. So that, a vote from wallet address A to wallet address B can be seen by anyone who has access to the chain. Which is, of course, a big disadvantage. And, it is not possible to use such a system for official/critical elections. Providing this anonymity is also a major challenge in the current state-of the-art works. Hao et al. in their work, proposed a solution based on the Diffie- Hellman process, which also implies the use of public/private key pairs and random numbers, so that a "two-round" referendum can supposedly be held with some ballot privacy [12]

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025
DOI: 10.17148/IJARCCE.2025.14347

IV. CONCLUSION

In this paper, we presented a novel blockchain-based electronic voting system that uses smart contracts to ensure voter privacy while facilitating a safe and economical election. In contrast to earlier research, we have demonstrated that blockchain technology presents a fresh opportunity for democratic nations to transition from paper-based elections to more efficient ones that save money and time. Additionally, we have improved the security measures of the current system and opened up new avenues for transparency. In both political and scientific circles, e-voting remains a contentious issue. Even though there are some excellent examples, the majority of which are still in use, Numerous other attempts either had major usability and scalability problems or failed to offer the security and privacy features of a traditional election [8].

On the contrary, blockchain-based electronic voting solutions, such as the one we have used with smart contracts and the Ethereum network, address (or may address with appropriate modifications) nearly all of the security issues, including voter privacy, vote integrity, vote verification and non-repudiation, and counting transparency.

However, there are some characteristics that the blockchain alone cannot handle. For instance, voter authentication (personal, not account-level) necessitates the integration of other mechanisms, like the use of biometric factors. Blockchain technology has a lot of promise, but in its current state, it requires a lot more research and currently might not reach its full potential. To enhance its support for increasingly sophisticated applications, the fundamental blockchain technology requires a concentrated effort.

REFERENCES

[1]. S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system”, [Online]. Available: https://bitcoin.org/bitcoin.pdf .
[2]. Ali Kaan Koç, Emre Yavuz, Umut Can Çabuk, Gökhan Dalkılıç “Towards Secure E-Voting Using Ethereum Blockchain”
[3]. G. Wood, "Ethereum: a secure decentralised generalised transaction ledger", Ethereum Project Yellow Paper, vol. 151,
[4]. pp. 1-32, 2014.
[5]. C.D. Clack, V.A. Bakshi, and L. Braine, “Smart contract templates: foundations, design landscape and research directions”, Mar 2017, arXiv:1608.00771.
[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/profile/Umut_Cabuk/ publication/308796230_E- Democracy_The_Next_Generation_ Direct_Democracy_and_Applicability_in_Turkey/links/581 8a6d408 a ee7cdc685b40b/E- Democracy-The-Next- Generation DirectDemocracy-and-Applicability-in- Turkey.pdf.
[8]. "Final report: study on eGovernment and the reduction of administrative burden (SMART 2012/0061)", 2014, [Online]. Available: https://ec.europa.eu/digital- single market/en/news/finalreport-study-egovernment-and-reduction administrative-burdensmart-20120061
[9]. F. Hao and P.Y.A. Ryan, Real-World Electronic Voting: Design, Analysis and Deployment, CRC Press, pp. 143-170, 2017.
[10]. 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.
[11]. Nir Kshetri, Jeffrey Voas, “Blockchain-Enabled E- Voting”.
[12]. 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. [12] U.C. Çabuk, T. Şenocak, E. Demir, and A. Çavdar, “A Proposal on initial remote user enrollment for IVR-based voice authentication systems”, Int. J. of Advanced Research in Computer and Communication Engineering, vol 6, pp.118- 123, July 2017. [13] Y. Takabatake, D. Kotani, and Y. Okabe, “An anonymous distributed electronic voting system using Zerocoin“, IEICE Technical Report, pp. 127-131, 2016.