

Voter Whitelisting in Blockchain-based E-Voting using Merkle Tree

Ishita Bhardwaj¹, Vani Chitkara², Megha³ and Richa Yadav⁴

¹⁻⁴Indira Gandhi Delhi Technical University for Women, Delhi, India

Email: {bhardwajishita15, vanichitkara18, m2001vidya}@gmail.com, richayadav@igdtuw.ac.in

Abstract—Blockchain-based e-voting systems use smart contracts to eliminate intermediaries, such as election officials, to automate voting. Traditional e-voting mechanisms have been demonstrated to be obscure and thus have led to declining voter confidence. This paper proposes deploying the voter application on blockchain for a decentralized environment to provide open and transparent voting. While ERC20 tokens diminish plural voting, Merkle Tree ensures scalable storage of voter data for whitelisting. The proposed solution can overcome scalability, double-spending, and plural voting challenges. The results documented in the paper prove that the proposed solution is efficient in the gas units spent and ensures faster verification than the existing solutions. Thus, the proposed blockchain e-voting mechanism ensures significant features of an ideal e-voting system.

Index Terms— E-Voting, Blockchain, Smart Contracts, Merkle Tree, ERC20.

I. INTRODUCTION

Over the past two decades, a rapid evolution has occurred in how people being part of significant institutions of democracy or any institution have been empowered by innovation in voting mechanisms. The development of digital technology has opened up opportunities for voters to express their opinions and elect governing bodies fairly and openly. With the technological transition from tradition-al ballot paper to electronic voting, precise and efficient criteria have been set to make elections more reliable. Electronic voting fulfils all the set criteria for a reliable electing process but fails to ensure concerns on anonymity, security and authenticated verification. The currently proposed e-voting mechanisms have demonstrated obscurity and a possibility of vote tampering, leading to a lack of confidence among voters.

Blockchain technology has garnered much attention in recent years. It has opened new avenues for implementing e-voting methodologies to deal with concerns regarding distrust in e-voting, authentication, anonymity, and centralised control, making the process trustworthy and reliable [1].

Blockchain voting is being proposed to eliminate the need for intermediaries required to conduct elections, such as election officials, by using smart contracts to automate the voting process. Eliminating any third party to conduct the elections ensures the non-tampering of votes and a fair vote count. The public dis-tributed blockchain ledger allows us to conduct just and secure elections that reinstate the fundamentals of democratic elections.

A. Challenges to Blockchain Voting Mechanism

Researchers have been drawn to blockchain technology and have undertaken several works to implement a

secure blockchain voting mechanism and integrate other emerging technologies. However, implementing Blockchain in Voting is debatable, given its various challenges.

- **Scalability:** Enough computational power is increasingly important to ensure a scalable voting platform for increased user and related blockchain transactions.[2]
- **Double-spending:** Double spending is a challenge where a voter can potentially cast multiple votes by sending multiple transactions with the same vote. The system needs to ensure that each vote is counted only once while maintaining the privacy of the voter's identity [3].
- **Slow transaction time and high transaction fees:** Scaling a blockchain application will lead to slow transaction processing times and high transaction fees, making it difficult to handle large volumes of votes.
- **Plural Voting:** Preventing Plural Voting is equally challenging since there are no secure means to verify a voter's identity without compromising anonymity. The non-availability of a security verification mechanism makes it easy for bad actors to exploit the system and cast plural votes.

B. Preliminaries

In the given section, a few technical terms have been explained that have been continuously used throughout the study and will help to understand how these techniques play an essential role in developing the proposed mechanism.

- 1) **Blockchain:** A blockchain is a decentralised ledger composed of blocks containing a set of transactions. The blocks in the chain are linked, creating a chronological data sequence. This linking mechanism makes the blockchain tamper-evident and secure, as any attempt to modify a single block would invalidate all subsequent blocks.[4]
- 2) **Merkle Tree:** A Merkle tree is a binary data structure used in cryptography and computer science to verify the integrity and authenticity of large data sets efficiently. [5]
A Merkle tree consists of a set of leaf nodes, each representing a piece of data, and intermediate nodes representing the hash values of their child nodes. In contrast, the entire set's hash value is represented using the root node. Recursive pairs of nodes are constructed until a single root hash is obtained. Hence the space complexity is $O(n)$. Therefore, Merkle trees enable efficient verification of large data sets by only verifying a small number of hash values, leading to a search time complexity of $O(\log n)$. [6]
- 3) **ERC20:** ERC20, or Ethereum Request for Comments 20, is a contract that keeps track of fungible tokens and regulates the minting, burning, transfer, approval, and total supply of coins. These contracts have a property that makes each token precisely the same (in type and value) as another. These tokens do not have special features or behaviour associated with them. Hence, they can be utilised as a medium of exchange currency for granting voting rights, staking, and much more. [7]

II. LITERATURE SURVEY

This section broadly discusses the previous work recorded to contribute to electronic voting. Multiple attempts have addressed the drawbacks of traditional e-voting mechanisms by proposing voting mechanisms built on the foundation of blockchain technology.

Motivated by recent advances, Abuidris et al. [8] proposed a blockchain-based e-voting system, allowing voters to vote using a smartphone or computer. Each voter receives a digital coin, which represents one opportunity to vote. The system uses sharding to improve scalability and performance. However, the authors also acknowledge potential drawbacks to this approach. For example, the system may be vulnerable to double-spending and Sybil attacks.

Additionally, there may be concerns about privacy and anonymity in a blockchain-based e-voting system. Merkle [9] provides the concept of a DAO Democracy, which combines the principles of Decentralized Autonomous Organizations (DAOs) with democratic decision-making. The proposed mechanism uses blockchain technology to create a decentralised system where citizens can vote on proposals and decisions. The paper argues that this approach would allow for more stable, less erratic decision-making that better reflects the needs and expertise of all citizens. However, the paper also acknowledges potential drawbacks to this approach. For example, there may be challenges in ensuring that all citizens have equal access to voting and decision-making processes. Concerns about the potential for manipulation or abuse of the system by certain groups or individuals.

Another blockchain e-voting mechanism provided by Giraldo, Barbosa and Gamboa [10] gives a proof of

concept that utilises smart contracts to conduct the elections. Their concept has multiple drawbacks, ranging from a lack of privacy measures to ensure the lack of encryption methods to safeguard the voters' and votes' data. Their mechanism needs to address the scalability issues of blockchain voting systems and the slow and pricey mechanism to conduct elections. Almas et al. [11] suggest a minimalistic blockchain voting mechanism that eases the election process. However, the system lacks privacy and security safeguards to ensure zero data breaches. The system needs to be more scalable, so nationwide elections cannot be conducted on their proposed system.

The blockchain e-voting system designed by Lee, Neo and Teo [12] encrypts the vote to ensure the voter's privacy by limiting the visibility of the voter's choice to the voter itself. However, the voter's identity is not precisely private since all the users use the same public key. In a study on the Merkle tree by Casellon et al.[13], the authors have attempted to propose a novel approach to reduce energy consumption by reducing computational overhead in a traditional Merkle tree using a two-level Merkle tree (TMT). The TMT binary structure has a root hash of the lower level in the upper level, while the lower level contains the hashed transaction as leaf nodes, which is expected to achieve energy savings of up to 80%. The authors also demonstrate applying the proposed approach in a real-world scenario. It is necessary to provide a detailed analysis of how the approach proposed affects security. In addition, the authors need to consider scalability issues in large-scale blockchain environments. Overall, the proposed approach of using a two-level Merkle tree to reduce energy consumption is an exciting idea that warrants further investigation. The study proposing a four-phased self-tallying blockchain-based e-voting system by Huang et al. [14] aims at ensuring maximum voter privacy while maintaining transparency and accuracy. The innovative features of self-tallying allow automatic vote count without the intervention of a centralised authority. The authors have used Zero Knowledge proof in the tallying phase.

While there are potential security threats to the proposed system, this four-phase voting mechanism, with setup, registration, voting, and tallying, is a secure, transparent, and anonymous voting system using blockchain technology. Table I. broadly compares a few more works incorporating blockchain into electronic voting systems based on the basic features expected in a blockchain system.

Researchers worldwide have proposed several solutions for ensuring security, anonymity and scalability in a blockchain-powered e-voting system ranging from simple, smart contracts applications to applying AI and ML and proposing advanced IoT systems. However, it is essential to conduct a detailed analysis of its security implications and scalability before it can be adopted in large-scale blockchain systems.

TABLE I. COMPARISON OF RELATED AND PROPOSED WORK

PROPERTIES	[15]	[16]	[17]	[18]	[19]
Anonymity	X	✓	X	✓	X
Integrity	X	X	X	✓	✓
Privacy	X	X	X	✓	✓
Security	✓	✓	X	✓	✓
Verifiability	X	X	X	X	X
Decentralisation	X	✓	X	✓	X
Authentication	✓	✓	X	X	✓

III. MAJOR CONTRIBUTION OF THE PAPER

The proposed study employs blockchain technology using ERC20 tokens and Merkle Trees to build an e-voting system for real-time voting applications at several democratic institutions. The suggested methods documented in our proposal can tackle the following issues:

- **Decentralisation:** The proposed system is suggested to be built on a public ledger blockchain to maintain decentralisation against the current centralised and biased voting systems. All the

transactions will be seen publicly through every single node, which minimises the risk of a centralised power controlling the voting.

- **Pseudonymity:** The application maintains secrecy regarding the voter's identity. It does not reveal the voter's details to identify the whereabouts of the voter, including their personal information.
- **Double Spending:** The use of ERC 20 tokens ensures that the voter cannot vote more than once, as the governance token issued to them will be burnt after the vote is given.
- **Plural Voting:** The technology does not permit another individual to vote in place of the identified voter because the biometric and voter ID of the voter are mapped. The voter's timeframe and status are tracked by the governance token that was provided. This tiered authentication makes sure there are no instances of voting in the plural.
- **Space Efficiency:** Merkle trees are used for voter data storage and whitelisting eligible voters. It saves storage space by efficiently verifying extensive data sets and checking a few hash values computationally. Hence increasing the number of voters will only mean that a small hash needs to be saved on the blockchain.

IV. PROPOSED MECHANISM

In this section, the architecture and algorithm of the proposed mechanism is elaborated upon. The architecture explains the smart contracts deployed in the Dapp that execute the registration voting process. The algorithm section elaborates on the registration, verification and voting phase.

A. Architecture

A blockchain-based e-voting system has been proposed that uses Merkle Tree, for which a Decentralised Application (Dapp) can be deployed on the Ethereum Blockchain to vote. The proposed architecture flow is described as shown in Fig. 1 below.

The Dapp will consist of a front end and a back end. The backend is the smart contracts hosted on the blockchain. The smart contracts and the front end can then be integrated using ReactJS libraries, Ether.js, and web3.js to fetch the blockchain functionality. A signer and provider is required (like MetaMask) that builds a bridge between the local node and the main net (or the testnet). The data can be stored in a decentralised (IPFS) or centralised database. The Dapp can be tested, debugged and deployed using Hardhat.

The proposed e-voting mechanism utilises four smart contracts:

- **GovernanceToken.sol** is used to mint Governance Tokens to demarcate eligible and unique voters. Later, the tokens are burnt to stop usage when the votes are cast. It imports the ERC20 contract. The Governance Token, in future, will be used to tally which voter has voted or not or is an invalid voter.
- **VoterRegistration.sol** is used to register a new voter. The application sends only the required information to the blockchain. It verifies the authenticity of the voter and does not allow the same voter to register twice.
- **VoterVerification.sol** is used to verify if the voter is registered correctly and is eligible to vote, using a Merkle Tree Root. It ensures that the voter currently voting is the same person who registered before. The verification is done just before the voter moves forward to vote. The function `verify()` in the contract is a view-only function, which means it does not cost us any gas price to verify the voters.
- **Voting.sol** is used for casting a vote. Here the voter chooses amongst the options provided and votes for their preference. The Governance Token is burnt when voting for a candidate, and the counter indicating the votes cast increases.

B. Algorithm

The voting process is divided into 3 phases; Registration, Verification, and Voting. The whole application assumes that every voter has a government-issued Voter ID, a device with a stable internet connection and prior knowledge of the given technology. The Voter IDs ensure that the voter checks the conditions for eligibility to vote. The Dapp is just a platform for voting that provides the interface for easy conduct of elections. Eligible voters must register on the platform to ensure a smooth transition from offline to online voting. This phase is done days before voting to ease the voting process. On voting day, to cast a vote, the voters need to prove that they are the person they registered as, are not an imposter, and are yet to vote.

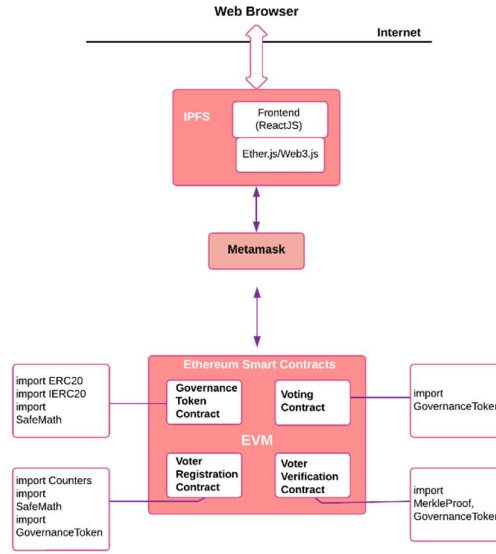


Figure 1. Architecture of the proposed voting mechanism

- Governance Token: In the application, to counter the problem of plural voting, all the eligible and new voters are given a Governance Token (GT), which will be minted in time. Each voter should have only 1GT, not more, not less.

$$GT = \begin{cases} 0 & ; \text{voter already voted or ineligible} \\ 1 & ; \text{voter is eligible to vote and has not voted} \\ > 1 & ; \text{voter is invalid} \end{cases}$$

- Registration Phase. The purpose of the registration phase is to collect the list of all eligible voters allowed to vote during the voting period. The voters must register using their government-provided Voter ID and biometrics, which will then be stored in the database. All the newly registered voters will receive a Governance Token as soon as they are successfully registered. Voter ID is needed to ensure that the voter is eligible, and biometrics is required to ensure that the registered person is also the person who will later be voting and not an imposter.
- Verification Phase. As soon as the registration phase is over, all the registered voters' IDs will be queried and compiled to build a Merkle Tree. The Merkle Root will then be sent to the contract. Merkle Tree is only formed when the voter registration phase is complete. Just before voting, the voter's Voter ID is verified against the Merkle Root for verification. Once verified, their input biometric is checked with the one mapped to the ID in the database. Ultimately, the voter's status is checked through their GT balance. It will tell if the person has already voted, has to vote or is an invalid voter.
- Voting Phase: As the voter is now verified, they land on the voting page where the list of candidates will be shown, and they will have to choose one amongst all the candidates. Once the voting is completed, the results will be published at the end.

V. PERFORMANCE EVALUATION AND COMPARISON

In this section, the proposed voting methodology is evaluated against the existing solutions to analyse its performance and record and interpret the results.

Table II. describes the functions in each smart contract and the gas units utilised, respectively. Note that the verify() method in VoterVerification.sol does not take any gas units as it is a view-only function. View functions do not cost gas if called externally outside the contract. However, they do cost gas if called internally by another function.

Table III. shows that the total gas units utilised to conduct elections through the proposed solution is 3,982,153. The calculated gas units of the proposed solution are almost five times less than those utilised by the blockchain

e-voting mechanism proposed by Pathak et al. [20], almost twelve times less than the solution proposed by Zhang et al. [21] and almost thirty-six times less than proposed by McCorry, Shahandashti and Hao in [2]. While deploying the Voting.sol contract, the candidate's data is used as the parameter. Hence the count of candidates increases whilst deploying the contract, leading to the gas units for the deployment increases.

As shown in Fig. 2., the gas units are increasing linearly with a constant change of 26,852 gas units. Please note, here, the gas units are of the deployed contract Voting.sol. Furthermore, the gas price can vary depending on the current price of ether on the side chains, testnets or mainnets. This leads to a linear function that predicts the gas units for deploying Voting.sol while adding x number of candidates.

On the other hand, the voter's information is sent to the already deployed contract, VoterRegistration.sol. The function register() sends the voter data to the contract individually. As the information sent to the contract is constant each time, the gas units for entering each voter are also constant. Hence the total gas units for entering x no. of voters will be linear.

On entering the first set of data, the gas units are seen to have a hike, and then as the voter information keeps on adding to the contract, the gas units become constant. Please note that the recorded gas units are of the function register(). Moreover, the records are not the same each time. Fig. 2b. showcases the pattern of gas units as the number of voters increases.

TABLE II. GAS UNITS UTILISED BY EACH SMART CONTRACT AND RESPECTIVE FUNCTIONS

CONTRACTS	Functions and the gas units used by them		
GovernanceToken.sol	deploy	mint()	burn()
	1,469,795	70,069	28,894
VoterRegistration.sol	deploy	register()	callGovernanceToken()
	723,583	44,099	261,121
VoterVerification.sol	deploy	setMerkleRoot()	verify()
	465,820	44,118	0
Voting.sol	deploy	vote()	callGovernanceToken()
	751,705	78,951	44,098

TABLE IV. COMPARISON OF TOTAL GAS UNIT UTILISATION BETWEEN RELATED AND PROPOSED WORK

PROPOSED SOLUTION	Ours	[2]	[20]	[21]
Gas units	3,982,15	145,381,858	21,225,237	47,160,611

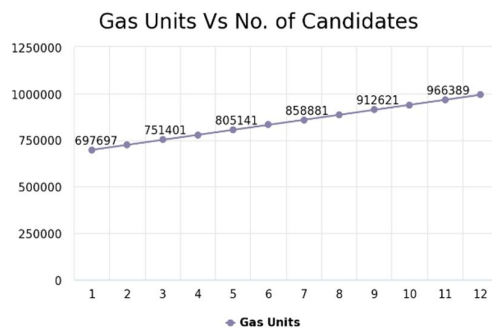


Figure 2a. Gas consumption for deploying Voting.sol each time the number of candidates increases

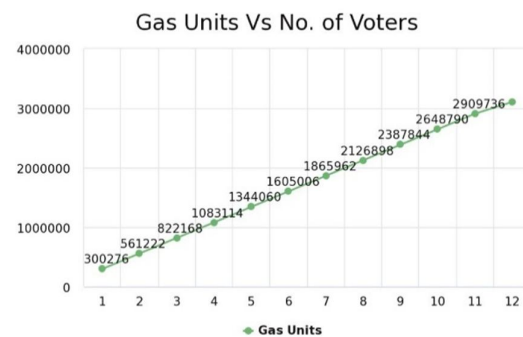


Fig. 2b. Total gas consumption of function register() each time a new voter is added to the system

VI. FUTURE SCOPE

In future work, further efforts will be made to make the proposed system more robust and decentralised. The current scenario of elections and voting mechanisms calls for a blend of centralised and decentralised applications. Complete decentralisation is not a practical approach for blockchain voting applications due to the voter and vote verification constraints. In order to make the proposed solution ready for practical use for conducting nationwide elections, government support will be required to verify voter IDs or Aadhaar IDs to verify the voters and check their eligibility to vote. The future work will aim to upgrade the proposed solution and achieve a more decentralised, secure, efficient and accessible system.

REFERENCES

- [1] M. Pawlak and A. Poniszewska-Marañda, "Trends in blockchain-based electronic voting systems," *Information Processing & Management*, 10-Apr-2021. [Online]. doi: 10.1016/j.ipm.2021.102595.
- [2] P. McCorry, S. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *Financial Cryptography and Data Security*. Sliema, Malta: Springer, 2017, pp. 357–375, doi: 10.1007/978-3-319-70972-7_20.
- [3] N. Kshetri and J. Voas, "Blockchain-Enabled E-Voting", *IEEE Software*, vol. 35, no. 4, pp. 95-99, 2018. doi: 10.1109/ms.2018.2801546
- [4] M. S. Farooq, U. Iftikhar, and A. Khelifi, "A framework to make voting system transparent using blockchain technology," *IEEE Access*, vol. 10, pp. 59959–59969, 2022. doi: 10.1109/ACCESS.2022.3180168
- [5] H. Liu, X. Luo, H. Liu and X. Xia, "Merkle Tree: A Fundamental Component of Blockchains," 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), Changchun, China, 2021, pp. 556-561, doi: 10.1109/EIECS53707.2021.9588047.
- [6] S. Jing, X. Zheng and Z. Chen, "Review and Investigation of Merkle Tree's Technical Principles and Related Application Fields," 2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), Xi'an, China, 2021, pp. 86-90, doi: 10.1109/CAIBDA53561.2021.00026.
- [7] K. H. Ansari and U. Kulkarni, "Implementation of ethereum request for comment (ERC20) token," SSRN, 27-May-2020. [Online]. doi: 10.2139/ssrn.3561395.
- [8] Abuidris, Y., Kumar, R., Yang, T. and Onginjo, J. (2021), Secure large-scale E-voting system based on blockchain contract using a hybrid consensus model combined with sharding. *ETRI Journal*, 43: 357-370. doi: 10.4218/etrij.2019-0362
- [9] R. Merkle, "DAOs, Democracy and Governance", *Cryonics Magazine*, vol. 37, no. 4, pp. 28–40, 2021.
- [10] F. Giraldo, B. Milton C. and C. Gamboa, "Electronic Voting Using Blockchain And Smart Contracts: Proof Of Concept", *IEEE Latin America Transactions*, vol. 18, no. 10, pp. 1743-1751, 2020. doi: 10.1109/tla.2020.9387645
- [11] M. Almas, E-Voting System using Smart Contracts and Blockchain (June 24, 2022). *International Journal of Innovative Research in Computer and Communication Engineering* | e-ISSN: 2320-9801, p-ISSN: 2320-9798 | Volume 10, Issue 6, June 2022 |, Available at SSRN: <https://ssrn.com/abstract=4228522>
- [12] C.-H. Lee, H.-F. Neo, and C.-C. Teo, "Secure e-voting system based on Blockchain technology," *Journal of System and Management Sciences*, 2022. [Online]. doi: 10.33168/JSMS.2022.0508.
- [13] C. Castellon, S. Roy, P. Kreidl, A. Dutta and L. Bölöni, "Energy Efficient Merkle Trees for Blockchains," 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 2021, pp. 1093-1099, doi: 10.1109/TrustCom53373.2021.00149.
- [14] J. Huang, D. He, Y. Chen, M. K. Khan and M. Luo, "A Blockchain-Based Self-Tallying Voting Protocol With Maximum Voter Privacy," in *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3808-3820, 1 Sept.-Oct. 2022, doi: 10.1109/TNSE.2022.3190909.
- [15] K. Srikrishnaswetha, S. Kumar, and M. Rashid Mahmood, "A study on Smart Electronics voting machine using face recognition and Aadhar verification with IOT," SpringerLink, 08-Feb-2019. [Online]. doi: 10.1007/978-981-13-3765-9_10.
- [16] R. Krishnamurthy, G. Rathee, and N. Jaglan, "An enhanced security mechanism through blockchain for E-polling/counting process using IOT devices - wireless networks," SpringerLink, 10-Aug-2019. [Online]. doi: 10.1007/s11276-019-02112-5.
- [17] X. I. Selvarani, M. Shruthi, R. Geethanjali, R. Syamala and S. Pavithra, "Secure voting system through SMS and using smart phone application," 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), Chennai, India, 2017, pp. 1-3, doi: 10.1109/ICAMMAET.2017.8186724.
- [18] T. Abayomi-Zannu, I. Odun-Ayo, and B. Fori, "A Proposed Mobile Voting Framework Utilising Blockchain Technology and Multi-Factor Authentication," *Journal of Physics: Conference Series*, 2019. [Online]. doi: 10.1088/1742-6596/1378/3/032104.
- [19] S. Ibrahim, M. Kamat, M. Salleh and S. R. A. Aziz, "Secure E-voting with blind signature," 4th National Conference of Telecommunication Technology, 2003. NCTT 2003 Proceedings., Shah Alam, Malaysia, 2003, pp. 193-197, doi: 10.1109/NCTT.2003.118833

- [20] Prof. M. Pathak, A. Suradkar, A. Kadam, A. Ghodeswar, and P. Parde, "A review on Blockchain based e-voting system," International Journal of Scientific Research in Science and Technology, 30-May-2021. [Online]. doi: 10.32628/IJSRST2182120
- [21] Y. Zhang, Y. Li, L. Fang, P. Chen and X. Dong, "Privacy-protected Electronic Voting System Based on Blockchin and Trusted Execution Environment," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 2019, pp. 1252-1257, doi: 10.1109/ICCC47050.2019.9064387.