e-ISSN: 3027-0650
Vol. 2, Issue 1, pp240-245, November 6 -9, 2024

Voter authentication Framework using Blockchain, Lightweight ring-neighbor-based user authentication and group-key agreement

Oluwatobi Balogun[1] and Abiodun Ogunseye[2]

{tabalogun@bellsuniversity.edu.ng[1], aaogunseye@oauife.edu.ng[2]}
Bells University of Technology, Ota[1], Obafemi Awolowo University, Ile-Ife, Osun State, Nigeria

Abstract

Global mistrust in election systems, caused by issues like vote rigging and hacking of electronic voting machines (EVMs), threatens democracy. This paper presents a blockchain-based e-voting framework designed to enhance voter authentication and secure voting transactions while addressing the computational challenges of traditional blockchain systems. The proposed system employs group-based voter authentication using a modified ring neighbour algorithm, allowing for decentralized voting, tamper-proof records, and the integrity of vote counts.

Key cryptographic elements such as SHA-256 hashing, Merkle trees, and random integer pairwise verification are integrated to secure the election process. A centralized Trusted Authority (TA) handles voter registration and block creation, preventing double voting and impersonation. The blockchain's immutability guarantees that any tampering attempts are computationally infeasible, ensuring secure and transparent election outcomes. This framework addresses the vulnerabilities of current electoral systems, combining cryptographic security, decentralized validation, and efficient voter authentication to ensure a reliable and secure e-voting process.

Keywords: Agreement, Authentication, Blockchain, Electronic voting, lightweight, validation

1. Introduction

The widespread mistrust that large segments of society around the world have for election systems poses a serious threat to democracy. Even the most developed democracies in the world, including the Republic of India, Japan, and the United States, still have imperfect legal systems. Booth capturing, election manipulation, vote rigging, and hacking of the electronic voting machine (EVM) are the primary problems with the current electoral system [1].

Electronic voting, or e-voting, has been around since the 1970s in a variety of configurations. Although securing the process has proven difficult, it offers fundamental advantages over paper-based systems, such as increased efficiency and decreased errors [2]. With the help of blockchain technology, networks can now store and distribute information without requiring external certification authorities or participant knowledge of one another [3]. Thus, blockchain has been used more and more to stop unauthorized transactions in a range of industries thanks to its strong cryptographic foundations. E-voting provides anonymity, or the capacity to vote in secret. The advantages of blockchain to the overall process of voting remain unmatched but due to the drawback of being computationally expensive, there is a reluctance to its application. This paper therefore developed a new blockchain framework that incorporates an agreement module among voters within a block of the blockchain to facilitate voter authentication and transaction validity

2. Literature Review

Beyond its use in cryptocurrencies, blockchain technology is gaining greater traction and finding applications in the Internet of Things (IoT) space [4] . [3]demonstrated how blockchain technology can be used to enhance the more widely used password-based authentication technology, which includes PKI-based authentication technology, smart card authentication technology, technology based on biological characteristics, and technology based on biological characteristics. Modern blockchain platforms have been developed to help overcome limitations and offer valuable value for additional modern business uses and applications. As of 2024, the top three blockchain frameworks for these use cases are R3 Corda, Hyperledger, and Ethereum. The global Corda Network is defined by the set of guidelines, network specifications, and associated governance procedures that comprise the Corda platform, which includes the open-source Corda software project [5].

[2] created a system to facilitate a real-world voting application that supports ledger synchronization, e-voting management, security, and access control, the blockchain e-voting system makes use of cryptographic properties.

[6], based on their system advocated that an election system should forbid the linking of votes to specific voters, permit safe authentication through the use of identity verification services, and ensure transparency by assuring each voter that their vote was counted accurately and without compromising their privacy. It should also make it illegal for any third

party to influence a vote, prohibit any one party from controlling the tabulation of votes or the declaration of the election results, and only allow eligible voters to cast ballots. [7] demonstrated the significance of participants' ownership in the blockchain for the integrity of transactions. They proposed integrating an intermediary server with the main smart contract to solve the voter anonymity issue and create a semi-decentralized system.

[8] conducted a survey to learn about the various blockchain system types that can be used in various contexts. According to their analysis, most blockchain applications used in voting scenarios are permission and primarily use the Proof of Stake consensus (or, in certain situations, the Proof of Authority consensus). Different blockchains were used in different scenarios, and in some cases, each person was assigned their own blockchain. This is due to the computational expenses of a typical blockchain necessitating the need for a lightweight version that still has the ability to facilitate all the properties of a properly run election

[9] suggested a protocol with lower computational and communication costs for group key agreement security, privacy protection, and limited computing power for the Internet of Drones (IoD). Drones that collect sensitive private data are highly susceptible to physical interception and data tampering, particularly when multiple drones are deployed simultaneously for collaborative purposes. They proposed that group communication requirements cannot be met by lightweight computations, conventional schemes, or the limited storage of IoD devices.

Three steps made up the proposed scheme: user registration, group member authentication, and group key agreement. Every user must register with TA in order to use the application. User management falls under the purview of TA, who is also in charge of adding new users and deleting those who are not registered. After a user completes the registration process, TA gives them a special secret token. Their approach builds group keys using binary asymmetric polynomials and uses addition as the primary mathematical operation by following these steps:

Step 1: Every group member chooses an integer at random and announces it to the group.

Step 2: Using their broadcasted values, members compute a shared pairwise key with each other in the group. A function applied to shares of the members' tokens is used to derive this key.

Step 3: Using a cryptographic hash function that combines the pairwise key and the random value of the other member, the member determines an authentication value to confirm identity. The other member receives this value as an authentication response.

Step 4: Using the pairwise key, the receiving member compares the received value to a recomputed value. A member is considered legitimate if their values align; if not, they are considered fraudulent. Until every group member has received their authentication, this procedure is repeated. The procedures guarantee reciprocal member authentication by using cryptographic techniques.

Bivariate polynomial-based methods not only offered information-theoretic security and authentication but also incurred lower computational costs when compared to public-key and lattice-based operations, which have high computational costs.

It was argued that security and scalability are two of blockchain's main problems. Concerning scalability [4] and [10] showed that rational miners would prefer to side with network attackers, and that the colluding group will expand until it consists of a majority, creating a new issue with selfish mining.

Two strategies have been put forth to address the problem of data storage in a blockchain environment: divide the traditional block into two parts, a "microblock" for transaction storing and a "key block" for leader elections; remove outdated transaction records and store them in a new database transaction tree to create a lightweight blockchain version; and prohibit the use of lightweight clients to perform costly computations in place of continuously creating new blocks [11] and [12].

3. Methodology

This paper propses a computationally inexpensive blockchain approach to authenticate the validity of voters in an electronic voting environment by using a group key and lightweight ring neighbour algorithm adopted from [9] and finetuned for this work. This approach affords the legitimacy of vote counts, election outcomes, and the ability to vote from anywhere. The process is described below

i. The voting application serves as the Trusted Authority (TA) and acts as the custodian for creating blocks, grouping, registration and deregistration of voters in the framework and has a user interface for interaction with the voters. They can access the application using a mobile device or computer over the internet. Voters have to register with the TA to be assigned to a group.

ii. Voter registration details must be made up of at least two unique identification code like the social security number or national identification number and a phone number to eliminate the possibility of double registration by voters. These details must be validated by the TA which allows for account creation. voters can then vote for the candidate of their choice

iii. The TA generates the genesis block and subsequent blocks of the blockchain. Usually, each block has a (i) BlockID to uniquely identify the block (ii) transactions (iii) Merkle root which identifies all the transactions within a given block (there can be more than one transaction within a block), (iv)hash value generated, that is cryptographic hash generated (v)previous hash that is, the hash value from the previous block. Every single block stores a reference to the previous block (vi)nonce which means the number only used once often used within blockchain mining to get the right hashing difficulty value and (vii) Timestamp which is the time a new block is added to the blockchain [13]. The genesis block contains no data asides for the BlockID and a randomly generated nonce which the SHA-256 algorithm uses to create the hash for the block. In this framework, it is only the genesis block that makes use of the nonce property.

iv. Voters are split into n number of groups where members of each group must be equal to 4 but not greater than 100. In the event of the remaining voters being less than 4, a module for direct voting caters for this by the TA and anomalies due to the minuscule number. A token made up of the eight last characters of the previous hash is broadcasted to members of a group to authenticate their identity. After identity authentication, a 4-bit Merkle root value is randomly generated and assigned to the group transaction. Each member can only have one transaction in the entirety of the blockchain.

v. Each member is then requested to generate a random integer that will be broadcast to the closest member of the group based on the assigned voter ID given upon completion of registration. Each voter derives the hash function based on the concatenation of the Merkle root and generated random number to authenticate the inputted hash function value of the paired voter to again authenticate the voter and validate the voting transaction. This guarantees that voters within a block are paired and validate each other's authenticity. Failure to complete this task causes a pair to be deregistered and added to another block upon re-registering on the application

vi. The BlockID, merkle root, and timestamp are then concatenated and hashed to form the hash of the current block which will be linked to the next block. Figure 1 shows a diagrammatic representation of the framework

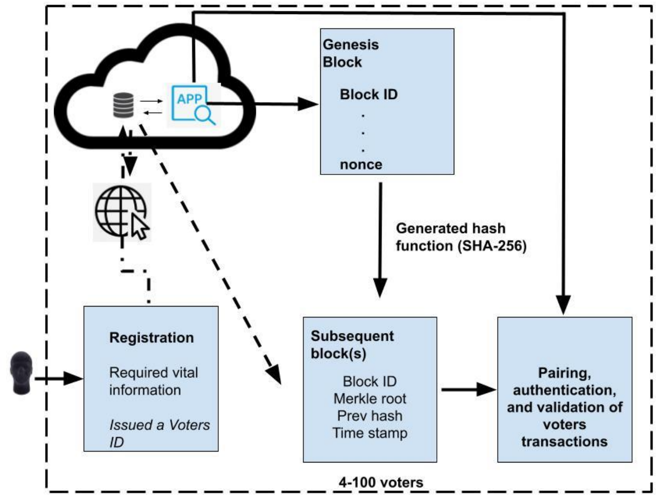

Fig 1. Blockchain-Based Voting System: Registration, Authentication, and Block Creation Workflow

4. Analysis:

a. Cryptographic Hash Functions (SHA-256) Function: A cryptographic hash function, such as SHA-256, takes an input and produces a fixed-size string of bytes. The output, called a hash, is typically represented as a hexadecimal number.

$$H(x) = SHA - 256(x) \quad (1)$$

Here, $H(x)$ is the hash value of the input xxx. The key property is that even a tiny change in x produces a significantly different $H(x)$, known as the "avalanche effect." Each block in the blockchain includes a hash of the previous block, H_{prev}, ensuring that altering any block would necessitate recalculating the hash for that block and all subsequent blocks:

$$H_{current} = SHA - 256(BlockID||Merkle\ root||\ Timestamp||H_{prev})(2)$$

This chaining mechanism makes the blockchain tamper-proof.

b. Merkle Trees: A Merkle tree is a binary tree where each leaf node represents the hash of a block of data, and each non-leaf node is the hash of its children. The root of the tree, known as the Merkle root, represents the entire dataset. If we have transactions $T_1, T_2, \dots, T_n$, the Merkle root M is computed as

$$M = H(H(T_1)||H(T_2)||H(T_3)||\dots||H(T_{n-1}) \qquad (3)$$

This construction allows for efficient and secure verification that a transaction is part of the blockchain. The Merkle root is used in each block to ensure the integrity of transactions. Altering any transaction would change the Merkle root, thus invalidating the block:

$$M_{current} = SHA - 256(Merkle\ root\ of\ transactions) \qquad (4)$$

c. Nonce: The nonce is a random or pseudo-random number added to the block's data before hashing. The goal is to find a nonce such that the hash of the block satisfies certain conditions (e.g., the hash starts with a certain number of zeros).

$$H(BlockID||Merkle\ root||\ Timestamp||H_{prev}) < Target$$

Finding such a nonce is computationally difficult, making it a proof-of-work system. The complexity arises because there is no shortcut; the only way to find a valid nonce is through trial and error. The genesis block uses the nonce property to generate the initial block's hash. This process ensures that even if an attacker tries to tamper with the block, they would need to find a new valid nonce, which is computationally infeasible.

d. Random Integer Generation and Pairwise Hash Verification: Each voter generates a random integer, which is shared with a paired voter. Both voters then use this random integer and the Merkle root to authenticate each other. Let r be the random integer generated by voter i and r_j the integer generated by their paired voter j. The hash used for verification is:

$$\boldsymbol{H_{i,j} = SHA - 256}(Merkle\ root||\boldsymbol{r_i}||\boldsymbol{r_j}) \quad (6)$$

Voter $\boldsymbol{i}$ verifies voter $\boldsymbol{j}$ by checking if:

$$\boldsymbol{H_{j,i} = SHA - 256(Merkle\ root}||\boldsymbol{r_j}||\boldsymbol{r_i}) \quad (7)$$

If the hashes match, the pair is verified. This pairwise verification ensures that each voter's identity and vote are validated by another voter, making it difficult for an attacker to forge votes without being detected. If an invalid or inconsistent hash is detected, the transaction is rejected.

Trusted Authority (TA) and Secure Registration

The TA acts as a centralized entity responsible for critical tasks such as voter registration, block creation, and voter authentication. Centralizing these tasks in a trusted and secure authority reduces the chances of unauthorized access or manipulation.

Voter registration requires the submission of at least two unique identification codes, such as a social security number and a phone number, which reduces the likelihood of double registration. The TA validates these details, ensuring that only legitimate voters are registered.

By creating secure accounts and associating them with unique identifiers, the system makes it difficult for attackers to impersonate or create fake identities.

The use of multiple unique identifiers makes it extremely difficult for an attacker to register multiple accounts, as they would need access to these identifiers for each account.

Centralized validation by the TA ensures that only genuine voters are allowed to participate, making unauthorized access nearly impossible.

Blockchain Structure

The blockchain structure used in the voting system ensures that each block contains unique BlockID, transactions, Merkle root, cryptographic hash, previous block's hash, and a timestamp.

The genesis block is the starting point of the blockchain, and it is uniquely identified by its BlockID and nonce, which are hashed using the SHA-256 algorithm.

Subsequent blocks are cryptographically linked, where each block contains the hash of the previous block, ensuring the integrity and immutability of the entire chain.

The cryptographic hash function (SHA-256) ensures that the content of each block cannot be altered without changing the hash. Since each block's hash is linked to the previous block, any attempt to tamper with a single block would require re-mining all subsequent blocks—a computationally infeasible task.

The Merkle root within each block further strengthens security by ensuring that any modification to a transaction within the block would change the Merkle root, thereby invalidating the block.

Since blocks are created by the TA, any unauthorized creation or modification of blocks by external or internal agents is highly unlikely.

Group-Based Voting and Authentication

Voters are split into groups, each containing 4 to 100 members. This grouping mechanism allows for efficient management of votes and ensures that each voter is authenticated by their peers within the group.

The use of an 8-character token (derived from the previous block’s hash) for group authentication further ensures that only legitimate group members can participate in the voting process.

A 4-bit Merkle root value is randomly generated for each group, ensuring that the transactions within the group are securely linked and validated.

The peer-to-peer authentication process within the group ensures that each voter’s identity is validated multiple times by different members, making it nearly impossible for an attacker to impersonate a voter.

The use of random tokens and Merkle roots adds an additional layer of security, as these values are unique and change with each transaction, making it difficult for an attacker to predict or reuse previous authentication data.

If any voter fails to authenticate correctly, the system automatically deregisters them, preventing any invalid transactions from being added to the blockchain.

Pairwise Authentication Using Hash Functions

After group authentication, each voter generates a random integer, which is shared with a paired voter. Each voter then derives a hash function based on the concatenation of the Merkle root and the generated random number. This process is used to authenticate the paired voter’s input.

The system ensures that each voter can only have one transaction within the blockchain, preventing double voting or tampering with the voting process.

The use of pairwise authentication ensures that each vote is securely validated by multiple parties. The hash function derived from the Merkle root and random number is unique, making it extremely difficult for an attacker to reverse-engineer or forge the hash.

The computational difficulty of generating valid hash values that match the required criteria adds another layer of security, ensuring that only legitimate transactions are recorded in the blockchain.

Immutable and Tamper-Proof Blockchain

Once a block is added to the blockchain, it is immutable due to the cryptographic linkage between blocks (hash of the previous block).

The BlockID, Merkle root, and timestamp are concatenated and hashed to form the current block’s hash, which is then linked to the next block. This chaining of blocks ensures the integrity and security of the entire voting process.

The immutability of the blockchain makes it resistant to tampering by both internal and external agents. Any attempt to modify a block would require altering all subsequent blocks, which would be computationally infeasible without controlling the majority of the network's hash power.

The use of secure cryptographic functions like SHA-256 ensures that even if an attacker gains access to the blockchain, they cannot alter the recorded data without detection.

5. Conclusion

The design of the system makes use of peer-to-peer authentication, cryptographic techniques, and the immutability of blockchain to produce a voting framework that is both highly secure and reliable. This system is immune to hacking attempts from both internal and external sources because it combines peer group authentication, centralized validation by the Trusted Authority, and the use of unique identification codes. The integrity and security of the voting process are preserved by the mathematical underpinnings of cryptographic hashing and the blockchain structure, which guarantee that any attempt to tamper with the system would be computationally impossible.

References

[1] Journal article: A. A. Lahane, J. P. T. P. and a. P. Potdar, "Blockchain technology based e-voting system.," in *ITM Web of Conferences*, 2020.

[2] Journal article: K. M. Khan, J. Arshad and M. M. Khan, "Secure Digital Voting System based on Blockchain Technology," 2018.

[3] Journal article: L. Liu and B. Xu, "Research on Information Security Technology Based on Blockchain," in *2018 the 3rd IEEE International Conference on Cloud Computing and Big Data Analysis*, 2018.

[4] Journal article: Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *2017 IEEE 6th International Congress on Big Data*, 2017.

[5] Website: G. Lawton, "Top 9 blockchain platforms to consider in 2024," TechTarget, 2024. [Online]. Available: https://www.techtarget.com/searchcio/feature/Top-9-blockchain-platforms-to-consider#:~:text=According%20to%20Menon%2C%20the%20top,and%20ConsenSys%20Quorum%20gaining%20ground..

[6] Journal article: F. Þ. Hjálmarsson, G. K. Hreiðarsson, M. Hamdaqa and G. Hjálmtýsson, "Blockchain-Based E-Voting System," in *2018 IEEE 11th International Conference on Cloud Computing*, 2018.

[7] Journal article: M. N. Neloy, M. A. Wahab, S. Wasif, A. A. Noman, M. Rahaman, T. H. Prant, A. K. M. B. Haque and R. M. Rahman, "A remote and cost-optimized voting system using blockchainand smart contract," pp. 1-17, 2022.

[8] Journal article: R. H. Sahib and E. S. Al-Shamery, "A Review on Distributed Blockchain Technology for E-voting," *Journal of Physics: Conference Series,* 2020.

[9] Journal article: Z. Zhao, C. Hsu, L. Harn4, Z. Xia, X. Jiang and a. L. Liu, "Lightweight ring-neighbor-based user authentication and group-key agreement for internet of drones," *Cybersecurity,* vol. 7, no. 50, 2024.

[10] Journal article: I. Eyal and E. G. Sirer, "Majority is not Enough: Bitcoin Mining is Vulnerable," vol. 61, no. 7, pp. 95-102, 2018.

[11] Website: J. Bruce, "The Mini-Blockchain Scheme," *2017.*

[12] Journal article: B. K. Kalejahi, J. Quluzad and S. Maharrəmli, "Development of Blockchain Technology," *J. ADV COMP ENG,* vol. 6, no. 4, pp. 265-272, 2020.

[13] Journal article: B. Oluwatobi, A. Oludele, O. Ernest and N. Uchenna, "Review of Blockchain Consensus Types, Applications and Drawbacks," *Research Journal of Mathematics and Computer Science,* vol. 2, no. 1, 2024.

[14] Journal article: L. Liu and B. Xu, "Research on information security technology based on blockchain," in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2018.