An Enhanced Blockchain-based Voting System Using Blake3 Cryptographic Hash Function

[1]Adepoju, J.A., [2]Olabiyisi S.O.[2] & [3]Ismaila W.O.
Department of Computer Science
Ladoke Akintola University of Technology
Ogbomoso, Oyo State, Nigeria
E-mails: [1] jaadepoju74@student.lautech.edu.ng; [2] soolabiyisi@lautech.edu.ng
[3]woismaila@lautech.edu.ng

ABSTRACT

This study developed a voting system based on the Solana blockchain which utilizes the Blake3 cryptographic hash function to enhance performance speed and program deployment cost. The blockchain-based voting system was designed using the Prototyping Approach and implemented using React.js was employed for the client side, Rust for the construction of Smart Contracts, Solana Playground a web-based IDE provided by Solana was used for the development and testing of Solana smart contracts, Chainstack for the connection of Solana blockchain network to the distributed application, and Phantom served as the crypto wallet for transacting on the distributed application. The Blake3-enabled distributed application was evaluated using speed and program deployment cost as perimeters. The developed system achieves a block time of 1.2 to 3.7 TPS, much more efficient than the existing system's 39 to 120 TPS, with a 97% reduction. Deployment costs in the developed system range from 0.0026 to 0.0169 Sol ($0.45 to $2.92), while the existing system costs 0.0023 to 0.0147 Ether ($8.10 to $51.78). With a conversion rate of 1 Sol = 0.0557 Ether, the developed system's deployment cost is 93.6% lower. The developed blockchain-based voting system improves performance over the existing system in terms of speed and program deployment cost making it a more appealing voting system for small-scale or large-scale applications.

Keywords: Blockchain, Voting System, Blake3 Cryptographic Hash Function, Security, Deployment

---

CISDI Journal Reference Format
Adepoju, J.A., Olabiyisi, S.O. & Ismaila W. O. (2024): An Enhanced Blockchain-based Voting System Using Blake3 Cryptographic Hash Function. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 15 No 3, Pp 1-14. dx.doi.org/10.22624/AIMS/CISDI/V15N3P1
Available online at www.isteams.net/cisdijournal

---

1. INTRODUCTION

The essence of voting lies in the collective influence of numbers. It serves as a crucial method for reaching collective decisions in various settings, be it within a group, meeting, gathering, or on a national scale. The process of voting ensures that each member's perspective is acknowledged and hinges on the majority to determine the outcome. Voting methods do not require detailed quantitative information as they rely on ordinal information, i.e. voters' expressed preference orders over the alternatives. This makes the decision process quick, transparent, and convenient to handle even in large groups [1]. In simpler terms, voters rank their choices in order of preference without assigning specific numerical values.

This ordinal ranking system simplifies the decision-making process, making it quick, transparent, and convenient, even when dealing with large groups of voters. The focus is on the relative preferences among options rather than precise numerical assessments. A society where the power is vested in the people or the general population is referred to as a democratic society. Voting stands as the foundation of democracy. Its evolution traces a compelling journey, showcasing the progress of democratic societies and the ongoing endeavour to guarantee elections that are fair, transparent, and accessible. Examining the evolution of voting systems, we have the following key systems such as Voice voting, Paper-based voting, Lever machine and punch cards, Electronic voting (E-Voting), Internet voting, Blockchain-based voting and Hybrid and mobile voting. Although the fascinating evolution experienced in e-voting has been coupled with a range of security apprehensions. While electronic voting systems offer the potential for increased accessibility and efficiency in elections, they also pose unique challenges related to the security, integrity, and privacy of the voting process.

Blockchain technology has gained widespread adoption globally, finding applications across various sectors of the economy, including cryptocurrency, media, entertainment, and notably, the voting system [2]. The integration of blockchain into e-voting, leveraging its inherent properties such as immutability, transparency, and decentralization, has conferred significant advantages. In the realm of blockchain-based voting, ensuring data integrity is a critical consideration, and cryptographic techniques play a pivotal role. Many implementations in this domain have traditionally relied on Secure Hash Algorithm-256 or Secure Hash Algorithm-3 cryptographic hash functions. However, the efficiency of these hash functions has been a notable concern. This research addresses the need for enhanced data integrity in blockchain-based systems by introducing the BLAKE3 cryptographic hash function. This choice aims to improve the overall effectiveness of cryptographic techniques employed in securing the integrity of voting data within the blockchain context.

In the contemporary era, our surroundings are undergoing rapid transformations [3]. According to technology and cybersecurity experts, the technological landscape is in a constant state of evolution, presenting boundless possibilities and advancements. However, this ongoing progress brings forth new challenges and threats, particularly in sectors like e-voting. Consequently, it becomes imperative to continually enhance systems to effectively address these emerging threats. This research introduces an advanced blockchain-based voting system that prioritizes security, reliability, and efficiency. The developed decentralized application accelerates the addition of blocks (sets of transactions) to the blockchain, thereby enhancing data integrity, scalability, and efficiency. These enhancements make the system well-suited for large-scale voting events.

This study presents the design and implementation of a secure electronic voting (e-voting) system leveraging the BLAKE3 cryptographic hash function. The system is developed and simulated using the Solana Command Line Interface tool, a blockchain platform known for its high-performance characteristics. The choice of BLAKE3 aims to enhance the security and integrity of the e-voting process, ensuring a robust and tamper-resistant foundation for electoral procedures. The study delves into the technical details of the smart contract implementation, emphasizing the integration of BLAKE3 and the simulation process facilitated by the Solana Command Line Interface tool. Through this research, we aim to contribute insights into the application of advanced cryptographic techniques in blockchain-based e-voting systems, with a focus on the Solana platform's capabilities.

2. RELATED WORKS

Based on smart contracts, one-time ring signatures and homomorphic encryption, [4] proposed a non-interactive and non-receipt electronic voting method. The smart contract is used to accomplish recording, managing, calculating and checking during the voting work. To protect the privacy of the electronic voting scheme a ring signature is employed to ensure that the registration and voting are anonymous. For large-scale voting in the blockchain, the paper proposed using DPoS for distributed consensus and miner nodes to randomize the votes and count the ballots.

[5] indicated that blockchain-supported voting systems may provide different solutions than traditional e-voting. They classified the main prevailing issues into the five following categories: general, integrity, coin-based, privacy and consensus. As a result of the research, it was determined that blockchain systems can provide solutions to certain problems that prevail in current election systems. On the other hand, privacy protection and transaction speed are the most frequently emphasized problems in blockchain applications. Security of remote participation and scalability should be improved for sustainable blockchain-based e-voting. It was then concluded that frameworks needed enhancements to be used in voting systems due to these reservations.

Several remote e-voting solutions based on blockchain technology have been proposed. Indeed, blockchain technology is proposed today as a new technical infrastructure for several types of IT applications because it allows the removal of the TTP and decentralizes transactions while offering transparent and fully protected data storage. In addition, it allows implementation in its environment of the smart contracts technology which is used to automate and execute agreements between users. [6] were interested in reviewing the most revealing e-voting solutions based on blockchain technology.

[7] used blockchain technology anonymity, mobility, integrity, security, and fairness in voting. By using blockchain their proposed system ensures security, privacy, and integrity. This system is proposed to provide voter anonymity by keeping the voter information as a hash in the blockchain. It also provides fairness by keeping the cast vote encrypted till the ending time of the election. After ending time, the voter can verify their cast vote, ensuring verifiability. In testing the protocol, Ethereum 2.0 a blockchain platform that uses Solidity as a programming language to create smart contracts. Smart contracts provide a safe means for performing voter verification, ensuring the correctness of voting results, making the counting system public, and protecting against fraudulent activities. They then analyzed the proposed system's performance based on security and gas costs.

[8] presented a paper that provides a study of hash functions comparison on holy Quran integrity verification to ensure the integrity of data in digital copy of holy Quran. it was reported that there are many methods available in the data security area for integrity verification and that the study presents a comparison of three cryptographic hash functions Secure Hash Algorithm 256, RIPEMD160 and Blake3 which are used to verify the integrity of the digital holy Quran and determine which one is better.

3. METHODOLOGY

During the development phase of the blockchain-based voting system, a range of software and hardware tools were employed to tackle the recognized challenges present in existing e-voting systems. These tools were meticulously chosen to optimize the system's speed and reduce the program deployment cost associated with the voting process. Throughout the development journey, these selected tools demonstrated their effectiveness in meeting the project objectives, notably by harnessing the strengths of the Solana blockchain and seamlessly integrating them with the React.js framework for the client-side interface.

The schematic depicted in Figure 1 delineates the involvement of two distinct entities: the administrator and the voters. The administrator is responsible for initiating elections and releasing results for completed ones. On the other hand, voters can peruse created elections and subsequently cast their votes. Both parties can access election results, whether the elections are ongoing or concluded. The architectural design for the Blockchain-based voting system is presented in Figure 1.

The developed system comprises six modules:

- Registration Module: This is responsible for initializing users either an administrator or a voter. The Voter Registration algorithm as shown in Algorithm 1, outlines the process of registering a new voter within the blockchain-based voting system. This algorithm ensures that each voter is uniquely identified, their registration status is updated, and their relevant information is stored securely. Input parameters are as follows:
- Admin Public key: The public key of the administrator overseeing the voter registration process.
- Voter Name: The name of the new voter.
- Vote Status: Indicates whether the voter has cast their vote or not.
- Voter ID: A unique identifier assigned to the new voter.
- Voter Public key: The public key associated with the new voter.
- Last Voter: The ID of the last registered voter, used to generate a unique voter ID.
- Voter Counter: Tracks the total number of registered voters.
  The algorithm gives an output of a voter account and a confirmation message indicating the successful registration of the new voter.
- Create Elections: This module encompasses the specification of election details, including the election name, index, candidate names, number of candidates, election results, and publication status, thereby placing the onus on the administrator. Algorithm 2 delineates the procedure for launching a fresh election within the blockchain-based voting system. It guarantees the precise and secure recording of essential election particulars, such as candidate details, voter status, and election outcomes.
- View Elections: Initialized voters can access a list of all elections created by the administrator and proceed to cast their votes for their preferred candidates.
- Vote: The system will maintain records of candidate information, and upon a voter casting their vote, it will be accurately recorded to ensure proper counting and adherence to the one-vote-per-voter principle. Algorithm 3 delineates the procedure for voters to cast their votes within the blockchain-based voting system. This algorithm guarantees precise recording of votes, updates election results accordingly, and signifies the voter's participation in the election.
- View Results: Voters and Administrators would be able to view the results of ongoing elections. Administrators' view results module would include the publish results functionality to achieve electoral results.

- Publish Results: After every election, a fresh block will be generated and appended to the blockchain, documenting the outcomes of the completed election in the publicly accessible result archive for future reference.

The View and Publish Result algorithm in Algorithm 4 outlines the process by which an administrator can view and publish the results of an election in the blockchain-based voting system. This algorithm ensures that only authorized administrators can access and publish the election results, and once published, the results are displayed for transparency. The algorithm displays the details of the election, including election name, ID, candidates, and any relevant information and also displays the results for each candidate, including the number of votes received. Mark the results as published to indicate that they have been made public. Return a success message indicating the successful publishing of the election results.

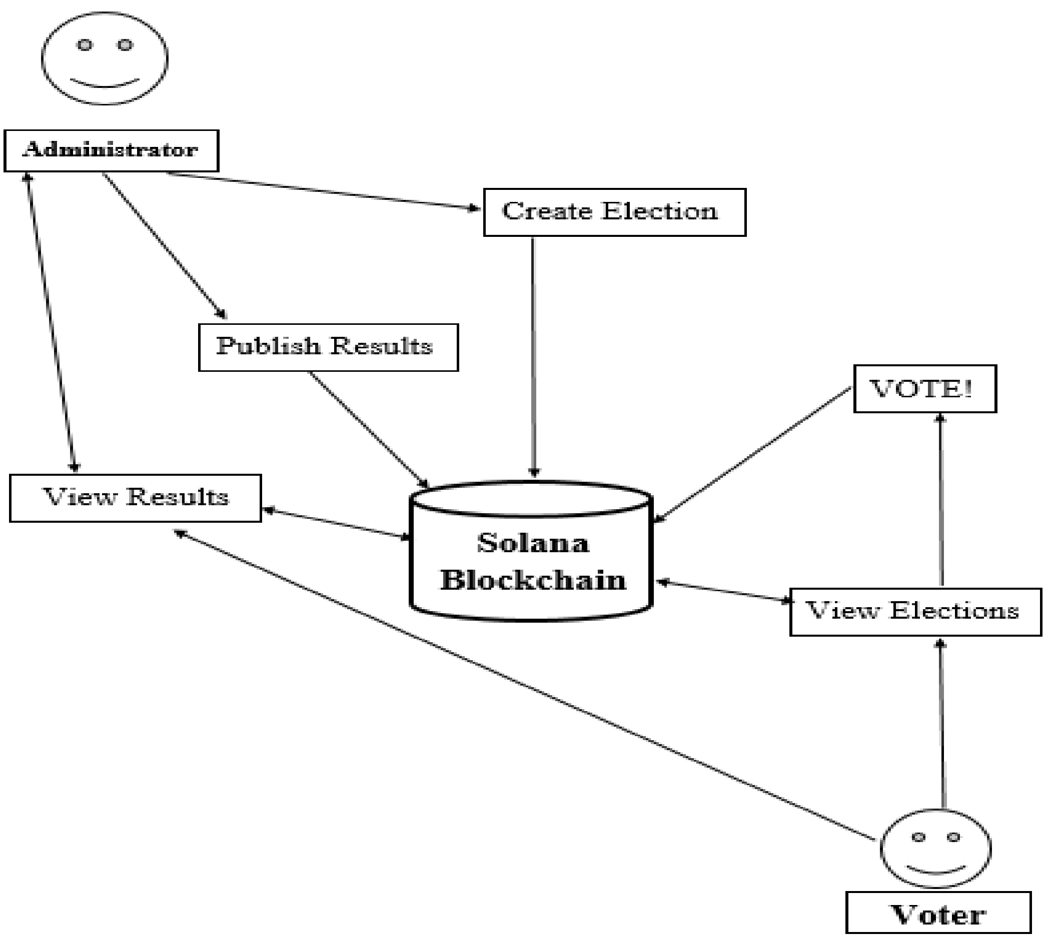

Figure 1: Architectural Design for the Blockchain-based Digital Voting System

Algorithm 1: Voter Registration

Input: Admin Public key, Voter Name, Vote Status, Voter ID, Voter Public key, Last Voter, Voter Counter
Output: Voter Account (Success Message)

1. Define VoterProfile Structure: Create a structure (VoterProfile) to hold voter-related data such as authority, voter ID, voter name, whether the voter has voted, registration status, and vote count.
2. Define RegisterVoter Structure: Create a structure (RegisterVoter) to represent the accounts needed for the register_voter function.
3. Input: Context with a RegisterVoter struct containing necessary accounts.
4. Retrieve the new_voter and admin accounts from the context.
5. Set the authority field of new_voter with the public key of the admin.
6. Assign a unique voter_id to the new voter based on the last_voter value of the admin.
7. Generate a default voter_name for the new voter.
8. Initialize the has_voted map for the new voter.
9. Set is_registered to true for the new voter.
10. Set the initial vote_count to 0 for the new voter.
11. Update last_voter and voter_counter in the admin account.
12. Return success.

Algorithm 2: Create Election

Input: Admin Public key, Number of Candidates, Candidates Names, Voters Status, Result, Result Status, Last Election, Election Counter
Output: Election Account (Success Message)

1. Define ElectionDetails Structure: Create a structure (ElectionDetails) to hold details about an election, including the admin, election name, election ID, election information, whether the result is published, voters who have voted, number of candidates, candidates' names, and election results.
2. Define CreateElections Structure: Create a structure (CreateElections) to represent the accounts needed for the create_election function.
3. Input: Context with a CreateElections struct containing necessary accounts. Parameters for the new election such as election name, election information, result publication status, etc.
4. Retrieve the new_election and admin accounts from the context.
5. Set the admin field of new_election with the public key of the admin.
6. Set other fields of new_election based on the provided parameters.
7. Initialize fields such as voter_voted, candidates_number, candidates_names, and results.
8. Print a message with information about the created election.
9. Update last_election and election_counter in the admin account.
10. Return success.

Algorithm 3: Vote Casting

Input: Voter Public key, Election Name, Candidates Names, Voter ID
Output: Updated Election Status (Success Message)

1. Input: ctx with Vote struct and vote parameters.
2. Retrieve the voter and election accounts from context.
3. Check if the voter has already cast a vote for the current election.
4. Else, cast the vote.
5. Update election results.
6. Mark the voter as having voted.
7. Update admin's counters for votes
8. Return success

Algorithm 4: View and Publish Result

Input: Admin Public key, Election Details

Output: Updated/Published Results (Success Message)

1. Input: ctx with ElectionDetails and AdminProfile accounts
2. Retrieve the election and admin accounts from context
3. Check if the caller is the admin
4. Check if the results have already been published
5. Display election details and results
6. Display results for each candidate
7. Mark the results as published
8. Return success

3.1 Technology Selection Criteria

The choice of a specific blockchain technology for this research depends on various factors, including project goals, technical requirements, and the features offered by different blockchain platforms.

3.1.1 Solana

While Ethereum is widely used and suitable for diverse applications, including voting systems, it currently lacks the Blake3 cryptographic hashing function. As a result, Solana, coupled with other technologies, was utilized to develop the distributed application. Solana is a high-performance blockchain network which was introduced in 2017 [9] and founded by Anatoly Yakovenko in March 2020 it was designed to provide fast and scalable decentralized applications (Dapps) and cryptocurrencies. Solana uses a unique consensus mechanism called Proof of History (PoH) in conjunction with a Proof of Stake (PoS) mechanism. PoH helps order transactions before they are processed in the PoS consensus, increasing overall network efficiency.

Solana is crafted for exceptional throughput, with a target of processing more than 50,000 transactions per second (TPS). Its scalability is achieved through a blend of innovative technologies, notably a unique timestamping approach, making it an ideal blockchain platform for this research. Solana prides itself on swift confirmation times, with block intervals of approximately 400 milliseconds, and it endeavours to offer cost-effective transactions to foster the creation of decentralized applications. Similar to Ethereum, Solana supports the programming of smart contracts using widely used languages such as Rust and C. The native cryptocurrency of the Solana network is “Sol”, utilized for staking, covering transaction fees, and engaging in governance activities. Solana is a public Blockchain platform created to overcome the scalability challenges that have plagued Blockchain technology [10]. Solana is crafted with a developer-friendly approach, offering support for diverse programming languages and detailed documentation.

Despite its growing popularity, Solana has encountered challenges like network outages and congestion, leading to continuous enhancements and optimizations. Given the ever-evolving landscape of blockchain projects, staying informed about the latest updates and developments from official sources is crucial for the most current information. Solana's pioneering consensus and scalability approach has established it as a prominent blockchain platform in the decentralized finance sector and beyond. Its rapid and efficient handling of transactions and smart contracts continues to draw attention.

Solana Command Line Interface which would be configured using devnet would be used for managing and testing the distributed application at the local machine. The Solana Command Line Interface comes with the “test validator” built in. To be able to compile the Rust-based Solana programs, Cargo would be installed alongside the Rust Programming Language. Cargo is also known as the Package Manager for Rust. Those would take care of the blockchain network and the backend technology but for connecting to the blockchain, performing transactions on the blockchain and the frontend would be handled by Chainstack, Phantom Wallet and React.js

3.1.2 Chainstack

Chainstack serves as a platform delivering infrastructure services tailored for blockchain developers. It presents a range of solutions designed to assist developers in the deployment and management of nodes across different blockchain networks. It allows developers to host nodes for multiple blockchain networks, including Ethereum, Binance Smart Chain, and others. Running a node is essential for interacting with a blockchain network, validating transactions, and accessing blockchain data. It offers API access to blockchain nodes. This allows developers to build applications that interact with blockchain networks programmatically.

3.1.3 Phantom Wallet

Phantom Wallet is a cryptocurrency wallet designed specifically for the Solana blockchain. Solana is a high-performance blockchain platform, and Phantom Wallet allows users to store, manage, and interact with Solana-based assets and decentralized applications (Dapps). It is a non-custodial wallet, meaning users have full control over their private keys. Phantom Wallet is dedicated to supporting Solana-based assets, including SOL (Solana's native cryptocurrency) and tokens built on the Solana blockchain. As a non-custodial wallet, it prioritizes security by allowing users to control their private keys. Users are responsible for securely storing and backing up their keys. Phantom Wallet is designed to work with various decentralized applications on the Solana blockchain, enabling users to interact with DeFi (Decentralized Finance) protocols, NFT (Non-Fungible Token) marketplaces, and other applications. Users can initiate and manage Solana transactions directly from the wallet. It often operates as a browser extension, seamlessly integrating with web browsers to provide easy access to Solana-based distributed applications.

3.1.4. React.js

React.js is an open-source JavaScript library for building user interfaces, particularly for single-page applications where UI updates are frequent. Developed and maintained by Facebook. React follows a component-based architecture, where the UI is broken down into reusable components. Components can be nested, composed, and managed independently, making the development process modular and scalable. It uses a virtual DOM to optimize the rendering process. Instead of directly manipulating the actual DOM, React creates a lightweight, in-memory representation of the DOM.

This allows React to update only the parts of the actual DOM that have changed, improving performance. React follows a unidirectional data flow, where data flows in a single direction from parent components to child components. This helps maintain a predictable state and simplifies debugging.

4. RESULT AND DISCUSSION

4.1 Result

The e-voting process was conducted through a distributed application deployed on the Solana blockchain network which leverages the efficiency of the Blake3 cryptographic hash function. This resulted in a notably swift response, manifesting in reduced response times compared to existing blockchain-based e-voting systems. The utilization of Blake3 ensures the system's data integrity, trustworthiness, and scalability. Gas usage is an essential aspect of blockchain technology, as it helps maintain the network's security and stability by incentivizing nodes to participate in the verification process. Additionally, the gas cost can be used to optimize transactions by minimizing the amount of gas spent, which can lead to cost savings for users and a more efficient network. Table 1 shows the program deployment costs which is an essential part of the gas cost for initial contract deployment and for different operations on the developed system using Blake3 (Solana) and an existing blockchain-based voting system (Ethereum) which uses Secure Hashing Algorithm 3 by [7], while Table 2 shows the block time for initial contract deployment and different operations provided by the developed blockchain based voting system (Solana) which uses Blake3 cryptographic hash function, result which was derived from Solana Playground (Solana Explorer) during testing and an existing blockchain-based voting system (Ethereum) which uses Secure Hashing Algorithm 3 by [7]. Block time refers to the approximate time it takes for a blockchain-based system to produce a new block, dictating the speed of transaction confirmation, which is measured in transactions per second (TPS).

Table 1: Program Deployment Cost for Developed System and [7]

Contract	Developed System		Existing System	
	Crypto Cost (SOL)	Actual Cost (Dollar)	Crypto Cost (Ether)	Actual Cost (Dollar)
Voter Contract	0.01414343	1.61	0.01228064	48.89
Candidate Contract	0.01335764	1.52	0.01159834	46.17
Voting Contract	0.01691487	1.93	0.01468706	58.47
Voter Registration	0.00263123	0.30	0.0022847	9.10
Candidate Registration (Election Creation)	0.00315410	0.36	0.0027387	10.90

Table 2: Block Time for Developed System and [7]

Contract	Block Time (TPS) Developed	Block Time (TPS) Existing
Voter Contract	3.7	120
Candidate Contract	3.4	110.5
Voting Contract	1.2	39
Voter Registration	2.5	81.5
Candidate Registration (Election Creation)	3.2	104

4.2 Discussion

The enhanced application using Blake3 cryptographic hashing function, as expected provides a faster block-generating platform. The system (smart contract) was simulated using Solana Playground, its results would depend on factors such as the system specification, the smart contract size and also the speed of the network.

4.3 Comparison of Enhanced Voting System and an Already Exciting System

In the current system, the actual expenditure in dollars for deploying contracts spans from $9.10 to $58.47. However, in the evolved system, the actual deployment costs range from approximately $0.30 to $10.90. This significant reduction in deployment expenses in the developed system, with an average reduction ratio of about 82%, underscores its cost-effectiveness compared to the existing system. Furthermore, the block time, indicative of transaction processing speed, in the current system ranges from 39 to 120 TPS. In contrast, the developed system boasts block times ranging from approximately 1.2 to 3.7 TPS. This substantial reduction in block time in the developed system, with an average reduction ratio of approximately 97%, highlights its superior efficiency compared to [7].

Based on the provided data, comparing the existing system (running on Ethereum) and the developed system (running on Solana) across various parameters. The developed system shows a significant reduction in gas usage across all contracts compared to the existing system. This reduction indicates improved efficiency and cost-effectiveness in terms of gas consumption. The crypto cost in SOL for the developed system is slightly higher than the corresponding costs in ETH for the existing system. However, this increase in cost is offset by the substantial reduction in gas usage, making the developed system more cost-effective overall. The developed system demonstrates significantly lower actual costs in dollars compared to [7]. This reduction in costs indicates improved affordability and accessibility for users in deploying contracts and participating in the system.

The existing system generally exhibits higher block times compared to the developed system. However, it's important to note that TPS alone may not fully reflect the performance and scalability of a blockchain system. Other factors such as network congestion, throughput, and consensus mechanism also play crucial roles. In summary, the developed system running on Solana demonstrates superior efficiency, cost-effectiveness, and faster transaction processing compared to the existing system, which operates on Ethereum. These improvements indicate the potential benefits of utilizing Solana's blockchain infrastructure (Blake3) for deploying decentralized applications.

Table 3: Comparison Summary

Evaluation	Ethereum (Secure Hashing Algorithm – 3)	Solana (Blake3)
Scalability	Very High	High
Block Time	Very Low	Low
Gas Fee	Very Low	Low
Transactions per Second	Very High	High

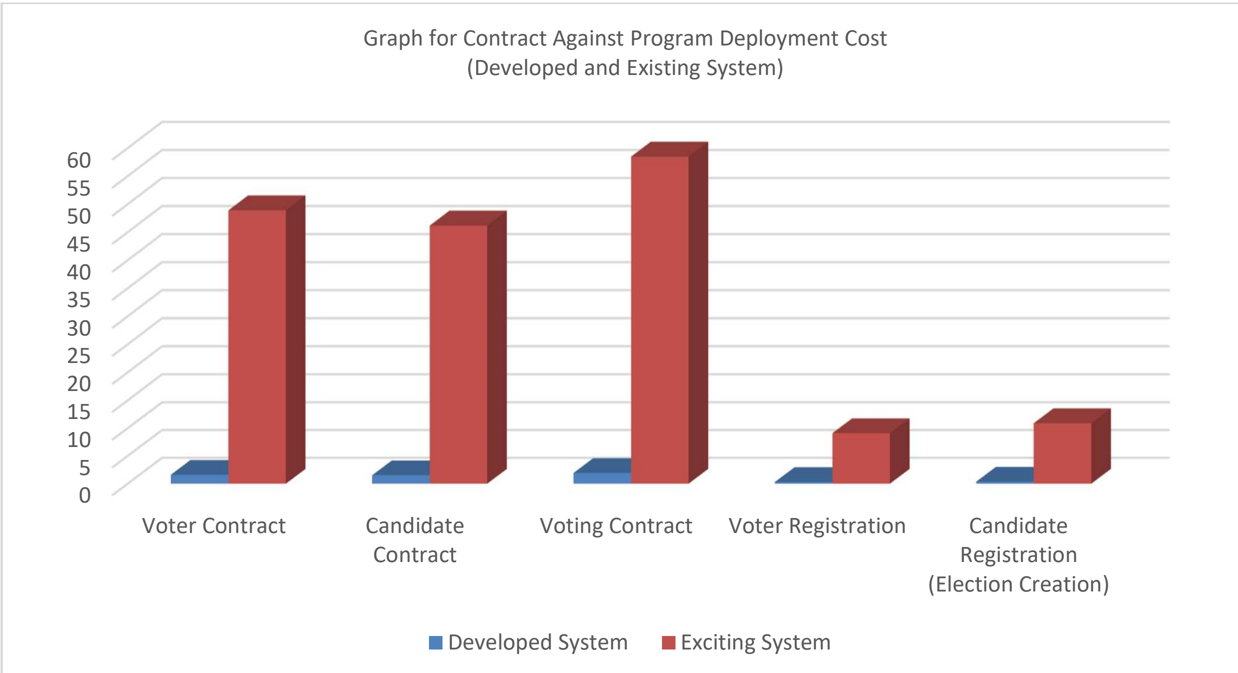

Figure 4: Comparison of Contract Deployment Cost.

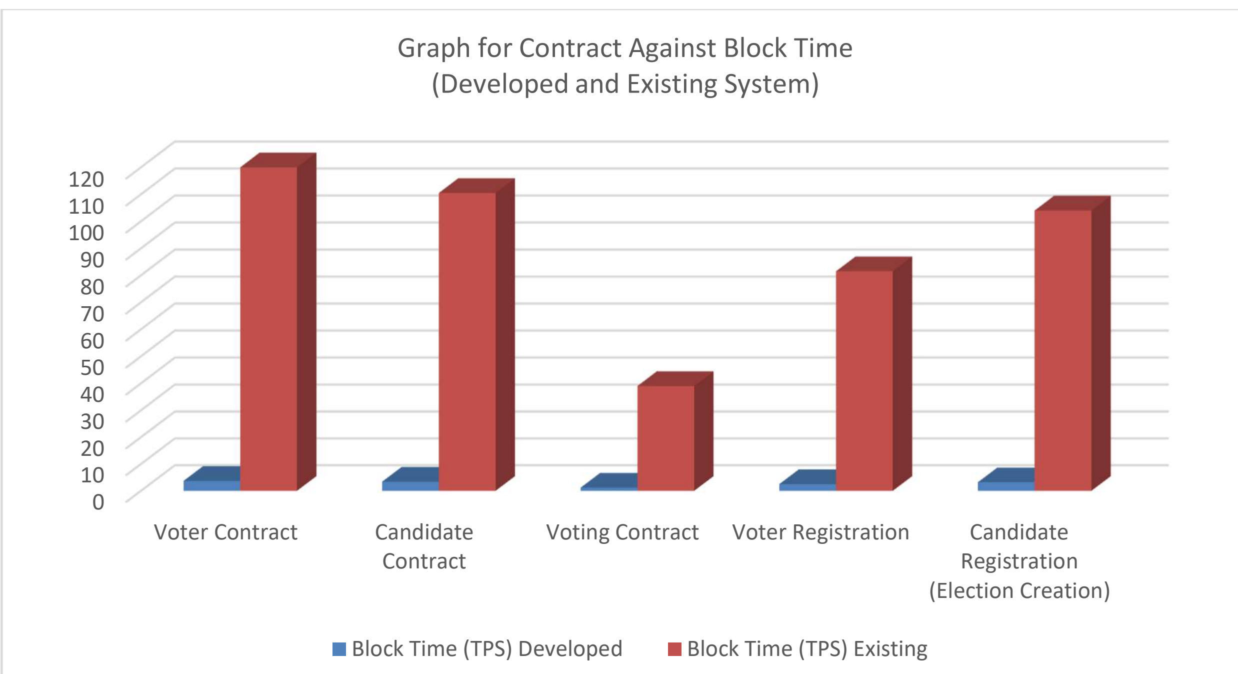

Figure 5: Comparison of Block Time.

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In conclusion, this research has undertaken a comprehensive exploration into the development and implementation of a blockchain-based e-voting system on the Solana blockchain network. The utilization of the Blake3 cryptographic hash function has not only contributed to achieving faster response times but has also significantly enhanced the system's security and overall efficiency. Through a distributed application, users can seamlessly engage in the e-voting process, ensuring a user-friendly experience.

This work has demonstrated the potential of blockchain technology, particularly on the Solana network, to address critical challenges associated with existing e-voting systems. The focus on data integrity, trustworthiness, and scalability aligns with the broader goals of ensuring a reliable and secure electoral process. The successful implementation of this blockchain-based e-voting system showcases its potential for application in real-world scenarios, providing an avenue for secure, transparent, and efficient electronic voting. As we move forward, continuous refinement and adaptation will be essential to keep pace with technological advancements and evolving security considerations in the realm of digital voting.

5.2 Recommendation

In light of the ever-evolving landscape of blockchain technology and its diverse applications, it is advisable to pursue additional research aimed at investigating emerging cryptographic hash functions and the latest developments in blockchain networks. This ongoing exploration is anticipated to play a crucial role in the enhancement of more resilient and secure e-voting systems. To affirm the scalability and security of the envisioned e-voting system, it is suggested to undertake real-world testing in cooperation with pertinent stakeholders, including election commissions, governmental entities, and cybersecurity specialists. Such collaborative efforts are poised to yield valuable insights into the system's functionality across a spectrum of scenarios, ensuring a comprehensive understanding of its performance.

5.3 Contribution to Knowledge

This research significantly contributes to the existing body of knowledge in the following ways:

- Enhanced security standards: The utilization of the Blake3 cryptographic hash function not only improves response times but also elevates the security standards of the entire e-voting ecosystem.
- Insights for stakeholders: The nuanced understanding and implementation of blockchain technology, particularly within the Solana framework, provide valuable insights for researchers, practitioners, and policymakers involved in electronic voting.
- Practical system architecture: The examination of the system's architecture, including authentication processes through the Phantom Wallet and the creation of elections by administrators, sheds light on practical aspects of implementing a secure and user-friendly e-voting system.
- Addressing key issues: By addressing critical issues such as data integrity, trustworthiness, and scalability, this research establishes a foundation for future endeavors in the domain of electronic voting.

REFERENCES

[1] Kangas, A., Leskinen, P. and Kangas, J. (2007) Comparison of Fuzzy and Statistical Approaches in Multicriteria Decision-Making. Forest Science, 53, 37-44.
[2] Techskill Brew, 21 Jan 2023, Blockchain Technology in Voting System, https://rejolut.com/blog/blockchain-technology-in-voting-system/
[3] Yash Soni, Leandros Maglaras, Mohamed Amine Ferrag, June 2020, Blockchain-Based Voting Systems
[4] Baocheng Wanga, Jiawei Suna, Yunhua Hea, Dandan Panga, Ningxiao Lua, Large-scale Election Based On Blockchain, North China University Of Technology, No.5 Jinyuanzhuang Road, Shijingshan District Beijing, Beijing 100144, China, 2018
[5] Ruhi Ta and Ömer Özgür Tanrıöver, A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting, Computer Engineering, Ankara University, 06830 Ankara, Turkey; ozgur.tanriover@ankara.edu.tr, Turkish Radio Television Corporation, IT Department, 06550 Ankara, Turkey, 9 August 2020
[6] Ali Benabdallah, Antoine Audras, Louis Coudert, Nour El Madhoun and Mohamad Badra, Analysis of Blockchain Solutions for E-Voting: A Systematic Literature Review, LIP6, CNRS, Sorbonne Université, 75005 Paris, France École pour l'Informatique et les Techniques Avancées (EPITA), 94270 Le Kremlin-Bicêtre, France College of Technological Innovation, Zayed University, Dubai, United Arab Emirates, 1 July 2022
[7] Syada Tasmia Alvi, Mohammed Nasir Uddin, Linta Islam, Sajib Ahamed, DVTChain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system, 2022
[8] Baqtian, H & Ali Al-Aidroos, Naziha. (2023). Three Hash Functions Comparison on Digital Holy Quran Integrity Verification. 11. 1-7.
[9] James Howell, How to build a dApp on Solana?, https://101blockchains.com/build-dApp-on-solana/, February 27, 2023
[10] TabTrader Team, What is Solana (SOL)?, https://tabtrader.com/academy/articles/what-is-solana-sol, Feb 17, 2023