International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *25 September 2023* ***Revised:*** *12 November 2023* ***Accepted:*** *30 November 2023*

Decentralize Internet Voting Model with Blockchain Smart Contract

Nnachi Lofty Amah[1], Ali Mansour [2]
Corresponding Author Email: loftdot@gmail.com
[1]School of Sciences, Department of Computer Science, Federal College of Education, Kontagora, Niger State, Nigeria. Email: loftdot@gmail.com
[2]Department of Computing and Information Systems, University of Bedfordshire, Luton, 3JU, United Kingdom. Email: ali.mansour@beds.ac.uk

Corresponding Author:Nnachi Lofty Amah,
Research scholar, School of Sciences, Department of Computer Science,
Federal College of Education, Kontagora Niger State, Nigeria.
Email: loftdot@gmail.com

Abstract: A blockchain smart contract for internet voting offers the benefits of decentralization, immutability, security, privacy, and transparency. The study adopts design science research approach to create and deploy a model utilizing Ethereum's platform, known for its flexible and expressive smart contract development capabilities. The study progresses through three stages: a review and contract implementation phase, a mining network phase, and a decentralized wallet application design phase. Using an Ethereum account, election authorities, candidates, and voters can generate a unique, encrypted identity key that can only be decrypted with a passphrase to conduct voting transactions. The system ensures real-time, verifiable election results, as it automatically closes at the designated deadline. The result prohibits a single voter from registering twice and voting for multiple candidates. The voting system utilizes the security features of blockchain technology to ensuring transparent, immutable, and secure voting transactions. Security and forensic experts can investigate the artifacts made of the blockchain technology's cryptographic algorithms. Future studies should evaluate the trade-offs between security and efficiency in different consensus mechanisms and optimization of blockchain resources for large scale elections.

Keywords*: Blockchain technology; decentralized voting, electoral security, ballot, smart contract, internet voting, electronic democracy*

1. Introduction

In this study, the potential of blockchain technology for decentralizing internet voting systems is explored. Both [1] and [2] underscore the importance of secure authentication and safeguarding ballot secrecy. Blockchain's potential in addressing these concerns is also emphasized. [3] and [4] dealt into the application of blockchain technology in electronic voting systems. While [3] emphasizing data security, [4] demonstrated an approach to system security, usability, and trust enhancement. [5] proposes a secure online voting system utilizing approaches like blockchain. More comprehensive evaluations of the blockchain-based voting system's scalability and effectiveness of use cases are necessary, according to the study. These studies emphasize the capability of blockchain technology to securely and reliably enhance internet voting systems, contributing to credible elections for democratic governance.

The Internet has evolved into what is known as the internet of values thanks, to technology. Blockchain is widely acknowledged as a system of distributed computing where network nodes carry out and maintain a log of all transactions grouped into blocks for reference [7], [8]. Because of the consensus among participants in the network it becomes tough to alter or undermine the system making it resistant to fraud and tampering. The debut of Bitcoin, in 2009 marked the introduction of a financial transaction system that offers security and faster transaction speeds [9], [10].

But in 2015, Ethereum gave rise to a new generation of smart contract platforms that offered a variety of decentralized autonomous organizations (DAOs) and decentralized applications (Dapps). It turns into a multifunctional tool with a wide range of end-user applications instead of just cryptocurrency. Pre-logical computation on a blockchain enables the smart contract to enforce itself automatically

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *25 September 2023* ***Revised:*** *12 November 2023* ***Accepted:*** *30 November 2023*

when a condition is met without the need for outside intervention. Units of value are transmitted at the time of code execution in a manner like that of data. Consequently, there is a shift from standardized contracts to standardized code and vice versa, as well as the capacity to develop a choice application using any rule that is specified in a contract [11].

This study is motivated by the need to build trust between the government and voters through an evolving, reliable, secure, and transparent electoral system. It hinges on design science and goes a step further by conducting more extensive performance evaluations of the blockchain-based voting system to assess its efficiency and effectiveness. This paper examines the blockchain technology frameworks and how smart contracts can be used to automate online voting. Hence, the research conceives the smart contract as a purposeful feature to set rules and conditions for electronic elections in a secure decentralized electoral infrastructure. This study identifies operational procedures and security measures specific to electronic voting and proposes a blockchain-based internet voting methodology. This methodology involves the implementation of both a decentralized Ethereum web and app wallet for voting transactions, followed by the implementation of a decentralized voting system to verify contractual voting transactions. To avoid duplicate registration and voting, the system makes use of computer power to validate transactions before adding them to the secret ledger. Voter anonymity and ballot security are also guaranteed. Evaluation of the result was accomplished with success. In use cases like cryptocurrencies, this emphasizes how crucial blockchain security and privacy are to preserving electronic democracy.

2.Literature Review

2.1 Blockchain Technology

Satoshi Nakamoto, who made the first presentation with the invention of Bitcoin cryptocurrency [9], presented the concept of blockchain. It is a cryptographically model of a distributed ledger of information shared among connected computers. The information represents transactions, contracts, deeds, assets, records, identities, or anything in digital form. Each node of the computer maintains a copy of the information and members must validate consensually any information and updates. Entry of new transactions is permanent, openly verifiable, and indestructible. Satoshi demonstrated the value of decentralized blockchain against the inherent flaws and monopoly in a centralized system wherein a single server could power massive applications, like storage and computing resources. The blockchain is worthwhile in that security, privacy, and trust become a value in a democratized distributed network. To understand how blockchain work, a discussion is made on figure 1, which presents attributes of core functionalities of the technology similarly illustrated in [12].

Decentralize Internet Voting Model with Blockchain Smart Contract

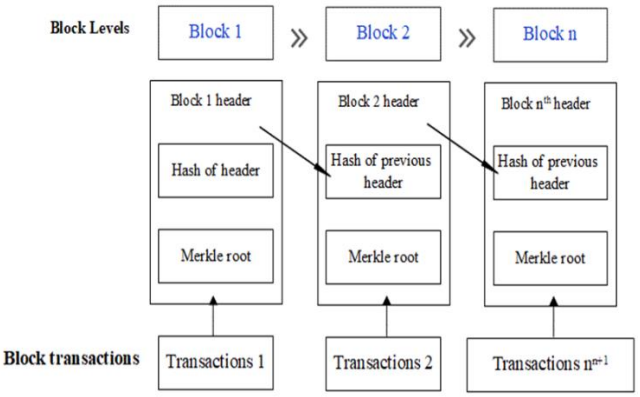

Fig. 1. Blockchain technology.

Advanced cryptography is used to secure each transaction in blockchain technology's decentralized ledger. Nodes of computers or servers in the network verify and consent to the validity of transactions based on established rules before they are added to the blockchain. Each data transaction in the block undergoes hashing, and a new hash is generated based on the previous hash and the current transaction data. With each update, a new block is appended to the chain at specified time intervals. The Merkle tree hash algorithm generates a single hash, termed the Merkle root, through the iterative hashing of paired hashes. In the blockchain, each block contains the hash of its predecessor, forming a sequential chain of interconnected hashes. The block's header includes a reference to the preceding block's hash along with transaction data, ensuring both security and interdependence. This interconnected structure is known as the blockchain. The blockchain network comprises numerous nodes, each utilizing an underlying algorithm to secure and preserve the integrity of records. These records, represented as transactions, document the history of state changes within the blockchain network [13], [14]. A blockchain is a continuous chain of blocks, each containing multiple transactions. Complex mathematical puzzles can only be solved using cryptographic hash functions with sufficient computational resources. Miners (computer nodes) were rewarded with cryptocurrency for solving the puzzle, and the block was distributed within an immutable and auditable blockchain network.

2.2 Adoption of blockchain technology

[15], [16] highlights the adoptions of blockchain technology to include the following benefits:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *25 September 2023* ***Revised:*** *12 November 2023* ***Accepted:*** *30 November 2023*

- Consensus verification: A blockchain decentralized system has no single system or point of authority to decide transactions. Nodes in the network consent and reserve copies of the agreement in a democratic consensus manner without disintermediation.
- Concept of immutability and anonymity: Records in blockchain are impervious to deletion, thereby fostering trust. The strong cryptographic hash ensures user anonymity, preventing other nodes from tracing personal details or linking transactions to interconnected nodes.
- No single point of failure: One system failure does not cause network-wide service interruption. The network's power comes from its interconnected, mutually dependent nodes, ensuring no system-wide failure point. In contrast to centralized systems, blockchain distributes power among the user base and eliminates the irritation caused by monopolistic service providers.
- Scalability: The decentralized system can adjust to increase the demand for computing power and user base with commensurate rewards in the form of cryptocurrency. The open-source posture guarantees the scalability of the platform to accommodate a community of users and innovations.
- Transparency and collaboration: Ease of sharing and visibility of transactions and the ability to initiate specific transactions after others have been finalized, in which collaboration is ensured without risk.

2.3 Security features of blockchain technology

The core tools of cryptography and data protection techniques, which employ coupled suites of algorithms to reach a strong consensus known as ‘proof of work’ have been the foundation for the blockchain's security, which has remained a mystery [17], [18]. The components of blockchain algorithms are highlighted:

1. Byzantine principle: In a dispersed network of nodes, [19], [20] investigate the situation where one node is the source sender, and the other nodes are peers. According to the Byzantine principle, in three different case scenarios, all fault-free nodes must concur on values from the sender independent of other faulty nodes. Initially, peers with no faults must agree on the same ideals. Second, the agreed-upon value's validity can only differ from that of the value sent by a flawless sender. At last, peers with no errors are in total accord. This principle involves no trust in the transaction and expectation of a high-level threat. This consensus algorithm uses three rounds of practical Byzantine fault tolerance (PBFT) to recheck that the blocks they will append are the same. The recipient must decrypt the message using the provided public key and then hash the original content. Utilizing ECDSA, users can prove ownership of a private key through digital signatures. In a trustless public Blockchain channel, a smaller key size reduces both storage needs and transmission requirements.
2. Asymmetric cryptography. An asymmetric pair of cryptographic keys is used by parties in blockchain platforms ([21], [22]). The recipient must decrypt the message using the provided public key and then hash the original content. Utilizing ECDSA, users can prove ownership of a private key through digital signatures. In a trustless public Blockchain channel, a smaller key size reduces both storage needs and transmission requirements.
3. Cryptographic hashing: [23] stress the application of the SHA-256 hash method and integrate the Merkle tree, which retains records of the previous transaction's hashes but contains compartments containing several hashes into one. A broadcast to network peers within a time stamp is made once the hashed transaction contents have been encrypted. Hashing algorithms also strengthen cryptographic encryption by making it impossible for private keys to undo a digitally signed hash back to its original state. Blockchain consensus verification and Proof of Work (PoW) are made easier by the SHA-256.
4. Peer-to-peer distributed computing: According to [24], [25], peer-to-peer computing is a distributed worldwide network of nodes operating without a central server whose job it is to collect, verify, and timestamp each transaction that occurs. As more transactions are added to the ledger, the nodes fight for the privilege of expanding it. It serves as a middleman, confirms transactions, and guarantees that fraud will never be tolerated or tampered with.

2.4 The Smart contract

Section 1 highlights how a smart contract functions according to the study. On the other hand, the general idea was initially presented in [26], [27], and was currently developed based on related studies. Smart contracts represent a paradigm shift toward complex scripting languages encoding self-executing agreements. These tamper-proof agreements are deployed on specialized blockchain platforms, facilitating the cryptographically secure signing and verifiable enforcement of contractual terms. Execution hinges upon the satisfaction of pre-defined conditions, as outlined in [28, 29]. However, this distributed approach necessitates the execution of contract code by all network nodes, potentially hindering scalability [32]. Consensus mechanisms play a critical role in securing these transactions within the blockchain, but their efficacy and potential vulnerabilities warrant further investigation [30]. Furthermore, smart contracts pave the way for the emergence of Decentralized Autonomous Organizations (DAOs). DAOs leverage smart contracts to encode organizational bylaws mathematically, fostering inherently democratic structures resistant to corruption due to the absence of a single point of failure [14, 31]. These

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *25 September 2023* ***Revised:*** *12 November 2023* ***Accepted:*** *30 November 2023*

autonomous entities operate based on pre-defined rules, eliminating the need for human intervention. However, the potential limitations of DAO governance models and their susceptibility to manipulation through complex code require further exploration [33].

2.4.1 Comparison of smart contract

A growing number of startups are emerging because of heightened buzz, and numerous open source blockchain systems are vying for user acceptance [32]. This calls for an awareness of programming patterns appropriate for creating prototypes of smart contracts. The severe results of a programming error and ineffective smart contract system resource management are demonstrated in [33].

The platform is a collection of devices and software that serve as a work environment for developers to create suitable software [34]. In certain areas of development demands, frameworks of reusable and already built software platforms can be called upon to support the platforms.

To highlight characteristics of popular flavors of smart contracts for the creation of decentralized applications, the research analyzes platforms in terms of design patterns [35], [36], [37], [38], [39], [40]. Table 1 shows a comparative review of selected smart contract platforms.

Table 1. Comparison of platforms for blockchain smart contract.

Platform	Ethereum	Hyperledger Fabric	Stellar	NEM
Application	Smart contract and cryptocurrency	Smart contract or chaincode	Smart contract, cryptocurrency	Smart asset, cryptocurrency
Operational Mode	Public and permissionless	Private and permissioned open blockchain	Public and Permissionless, but with the certainty of permissioned	Public or private and permissioned
Energy	High	Low	Low	Low
Smart Contract Execution	Ethereum Virtual Machine	Docker	Docker	Docker
Consensus	PoW and POS as	PBFT	FBA	PoI
Data Model	Transaction based	Key value	Account based	Transaction and account based
Transaction speed	High	low	low	low
Contract Language	Solidity, Serpent, LLL	Golang, Java, Kotlin	C/C++, JavaScript	Java
Specialty & advantage	Unlimited applications	Modular applications	Modular applications	Off-chain application and ease of development
Scalability	Existing scalability issue	Not prevalent	Existing scalability issue	Not prevalent
Privacy	Existing privacy issue	Not prevalent	Not prevalent	Not prevalent
Limitation	Likelihood of coding error	Limited distribution and coding error	Limited distribution	Less decentralized

Note: FBA: federated Byzantine agreement; PoI: proof of information; LLL: low-level Lisp-like language; PoW: proof of work; PoS: proof of stake; PBFT: practical Byzantine fault tolerance.

The proposed working prototype is developed on the Ethereum blockchain, based on the evaluation provided in table 1. With cross-platform implementation support, high processing power and fast contract execution, and countless wide-use applications, it has it all.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *25 September 2023* ***Revised:*** *12 November 2023* ***Accepted:*** *30 November 2023*

Stated differently, Ethereum adheres to an ideal protocol but has different regions that require implementation. It also has an EVM built in and is a general blockchain for creating smart contracts. As noted by [41], Ethereum's universality and scalability remain prospective even though human mistake inevitably leads to defects in scripts. It's a significant, trustworthy, and unexplored compute paradigm, with an extremely straightforward integrative platform for developers.

3. Methodology

The Ethereum blockchain's smart contract architecture is the focus of the suggested blockchain smart contract voting mechanism. The next step is to use web3 to create a wallet and decentralized application (Dapp) that will act as a conduit between voters, candidates, and election officials. Interacting with the Ethereum network of clients at the console is possible through the browser. After that, to be utilized for voting transactions via the web or application wallet, election authorities, candidates, and voters need to setup an Ethereum account that only produces key pairs that have been encrypted using the user's private key. Figure 2 depicts the envisioned Internet voting system's process diagram, which helps to conceptualize the prototype.

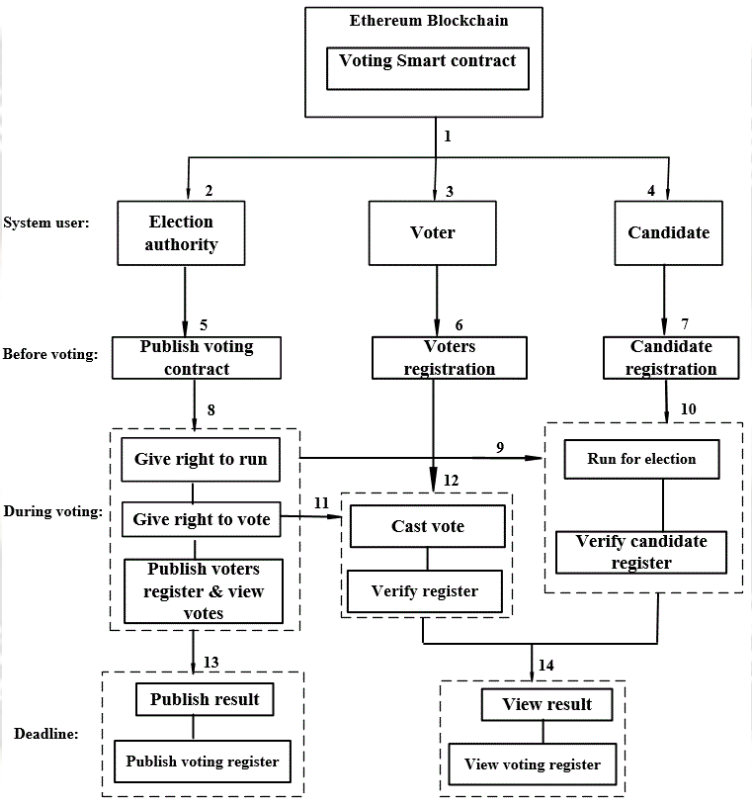

Fig. 2. Process chart of the proposed system

The program is modularized. It operates on the Ethereum Ropsten test network and interfaces with a development web server-based decentralized application (Dapp). Nonetheless, network nodes that are connected to one another maintain and update the Ethereum blockchain's database. The identical command is carried out by each network node that needs to run the EVM. The process of creating an account in any Ethereum wallet program, wherein the user inputs a password to encrypt the private key, is the first step in the acts taken by the election authority, candidates, and voters [45]. To enable transactions that can trigger computing codes of election rules encoded, the password and private key are utilized in the background to unlock the wallet.

As shown in figure 2, the events pertaining to the smart contract, election authority, candidates, and voters are described. It displays the architecture and connections among the Ethereum blockchain smart contract, private, bilateral, or multilateral agreements among voters that are recorded and invoked by events on or off the blockchain before, during, and post-voting actions of system users. The actions of the smart election contract prior to, during, and following the voting deadline are depicted in the stepwise scenario that follows.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *25 September 2023* ***Revised:*** *12 November 2023* ***Accepted:*** *30 November 2023*

1. The smart contract: Under electoral law, the election's rules, modalities, and guidelines are encoded into a smart contract, thereby establishing the election authority. The functions of the electoral authority, candidates, and voters are initially determined in the drafting of the election contract. The contract stipulates the election ballot format for computational rule abidance by transactional users, and events are initiated upon contract rule fulfillment. The electoral law binds stakeholders to activities before, during, and after the election. The election authority is granted administrative and legal authority to create the ballot and publish the voting smart contract, recognizing the rights of voters and candidates to participate. By ensuring transparency, the election process enables qualified individuals to participate and oversee the voting procedure. It ensures transparency of the voting process by enabling legitimate voters to confirm that their ballots are counted.
2. Election authority: In the Ethereum network, the smart contract's owner address functions as the election authority during transactions. The electoral authority draws its power from the law. The election authority authenticates registered candidates and voters for eligibility to run or vote. Single voters' trust would be bolstered through this addition.
3. Voter: The voter votes in accordance with contractual rules.
4. Candidate: In accordance with the contract rules, the candidate gives consent to run in the election.
5. Publish contract: The election authority publishes the Voting smart contract to the blockchain as the ballot. After publication of the contract, the election rules are fixed. The contract functions as the election authority while the election authority acts as the contract. The ballot is generated at contract publication and the election commences immediately thereafter, with a predetermined timeline and voting deadline.
6. Registration of voter: Using an Ethereum wallet address and a password to encrypt the private key, voters can verify their identities. Unlocking the wallet address for the voting transaction code requires both the password and a private key. No evidence of the voter's biological identification is left behind when using the related public key to generate the wallet address for their identity. restrictions on voter and candidate registration, as discussed in Section 1.5.
7. Registration of candidate: To create the candidate identity addresses, the above technique is done to candidate registration.
8. Electoral authority: The Election Authority provides the voter list and oversees an open vote count in addition to granting the ability to run for office and cast a ballot. The voting process is being validated in real time, and active involvement is permitted. Together with contract information, voter registration for candidates, and voting itself, election results are calculated. These and a plethora of other steps are combined to build the vote register.
9. Candidate receives right to run: Voters may cast ballots for the candidate who has been approved to run.
10. Activities of the candidate during voting: The candidate confirms the voting register and is granted verified right to receive vote.
11. Give the right to vote: Candidates may be chosen by voters who have obtained voter certification.
12. Activities of the voter during voting: Voters have access to the election register for verification. After choosing a candidate, the voter completes the ballot transaction, and the vote is counted.
13. Publish result: Results of the election are entirely transparent and are produced automatically as votes are cast. The election results are publicly available for open verification by the Election Authority.
14. Election results and voter registration are publicly accessible for candidates and voters to view and verify. According to the terms of the contract, voting closes.

3.1 Contract storage

An exact location for the smart contract is followed in the design. Contract storage can be organized with the use of Solidity's tools. The intelligent voting contract will thus have a distinct storage that it can write to and read from within the blockchain. Blocks comprised of transactions are connected to one another within the Ethereum network. Here, each transaction in the blockchain is stored within the set of blocks. Any of the mappings, structures, and state variables can be used as the contract storage specification or placeholder. There is also a key-value store, 2^256 potentially accessible keys, and 2^256 known value pairs in the contract storage. According to what has been said thus far, distributed nodes can verify signatures and avoid middlemen thanks to the blockchain's storage capacity [36]

3.1.1 Smart contract data storage

The following lists of tables provides an illustration of the storage specification for the smart contract's function parameters.

Table 2. Placeholder for election authority.

Name	Address	Value type	Value size
registerCandidate	candidateAddress	bytes	bytes32

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 September 2023 **Revised:** 12 November 2023 **Accepted:** 30 November 2023*

giveRightToVote	address voter	bytes	bytes32
giveRightToRun	address candidate	bytes	bytes32

Table 3. Placeholder for candidate registration.

Name	Address	Value type	Value size
registerCandidate	candidateAddress	bytes	bytes32

Table 4. Storage placeholder to register as a candidate.

Name	Address	Value type	Value size
registerVoter	voterAddress	bytes	bytes32
voteForCandidate	candidateAddress	bytes	bytes32

Table 5: Storage placeholder to register as a candidate

Name	Address	Value type	Value size
getVotes	candidateArray	bytes	bytes32
totalVotesFor	candidateAddress	bytes	bytes32
winnerName	winningCandidateAddress	bytes	bytes32

Table 6. Structs storage specification for a single voter.

Name	Value type	Value size
canVote	bool	bytes32
Voted	bool	bytes32

Table 7. Structs storage specification for a single candidate.

Name	Value type	Value size
RunningAddress	Constant	bytes32
canRun	bool	bytes32
voteCount	uint	bytes32

3.2 Verification and authentication process

Users' authentication depends on secp256k1 ECDSA for transaction signing and verification in a blockchain-based voting system that uses public key cryptography (PKC) [45]. A set of keys is in the possession of the election authority, which is used to digitally sign the voting transaction and validate it using the public key.

Crucially, the user's account address on their device communicates with the Ethereum network using a private key that powers an Ethereum node, and it does not save any identification information. The procedure for user verification and authentication goes like this: the election authority creates and initializes the voting process contract by generating cryptographic key pairs using a new Ethereum account and the public key as their address.

Subsequently, they distribute the public key through the Voting DApp URL after storing the contract on the blockchain associated with their address. In order to vote, a user must hash their account address using the Ethereumjs-util library's ecsign() function, which uses the public key of the authority to sign documents. The voter signs the contract and sends their vote, hashed address, and both. The erecover() function is used by the contract to get the user's address from the signature and to verify the vote by comparing it with the provided hash.

The function validates the signature and produces an error (value of 0) if it is not successful. Ultimately, the contract starts to implement the voting logic after it passes verification. Developed in a language intended to communicate with a backend, the voting DApp's front end and UI are browser-based applications. Swarm and IPFS, two distributed storage platforms for DApps, data, public records, and blockchain, are examples of decentralized storage locations where this front end is housed. Web3.js, which connects to a synchronized Ethereum network node and creates a blockchain data directory, is the interface via which communications with the Ethereum network take place.

To communicate with the blockchain via Remote Procedure Call (RPC), the voting contract is first transmitted from the front-end interface using the web3.js module. Communication between various processes on a single computer is made easier by the RPC client's encryption of communications, which also simulates the whole client behavior. Utilizing the Solidity compiler (solc), the contract is compiled and uploaded to the network, where it communicates with other contracts over the Ethereum web3.js JavaScript API.

The front end receives the built contract, which is now a .sol binary. To prevent transaction skipping, duplication, and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 September 2023 **Revised:** 12 November 2023 **Accepted:** 30 November 2023*

replay attacks, every call for a contract from the front end requires the exchange of an address key. The Ethereum nonce records the transaction completion and increases consecutively. The contract address and Application Binary Interface (ABI) are sent back to the front end after the process is finished. Later transactions are handled in a similar manner. The Web3 API can sign transactions before sending them to the Ethereum node, either on the front end or on the node itself. This includes transactions pertaining to the initiation.

4. Implementation

The various components of the smart contract code that make up the system implementation are synchronization, mining on the blockchain, private network implementation for Ethereum, address space for accounts and storage, and so on. Installing the Solidity program allowed users to write code segments for the smart voting contract's implementation. Additional tasks include wab3js function calls on underpinning blockchain algorithms for system security, as well as compilation, deployment, and migration. The blockchain Internet voting mechanism is created by successfully integrating various elements into a single system.

The next stages comprise the mining process and private network deployment for the system. The process of synchronization and mining in an Ethereum private network involves starting a peer-to-peer node and initializing the chain configuration. The disk is then enabled for Ethash, followed by the initialization of the Ethereum protocol and the loading of available blocks. Network ports are mapped, HTTP endpoints are opened, and the account address is unlocked. Finally, synchronization begins, fully integrating the node into the Ethereum private network as shown in figure 3.

This synchronization process imports new state entries, as indicated by the processed and pending entries in Figure 4. It kept importing fresh chain segments and building new blocks endlessly once the genesis synchronization was achieved. Next, the Ethereum test network's real-time mining operation and validation of transactions are depicted in Figure 5. The blockchain technology's algorithm is illustrated in this picture. It consists of multiple hash-based data transactions and periodic hash pairs up until the last single hash. This will establish a continuous chain link to the subsequent block and its hashes. This attests to the blockchain network's and the proposed voting contract's indestructibility and immutability.

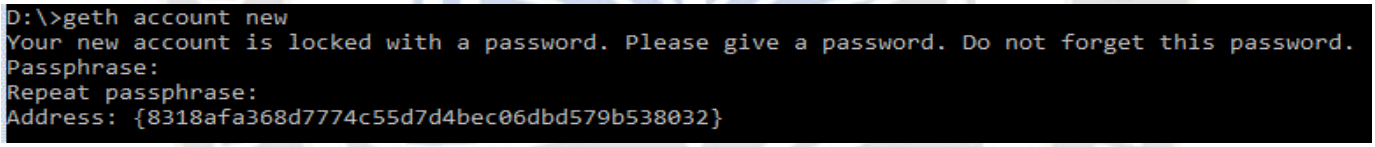

Fig. 3: Mining for client account.

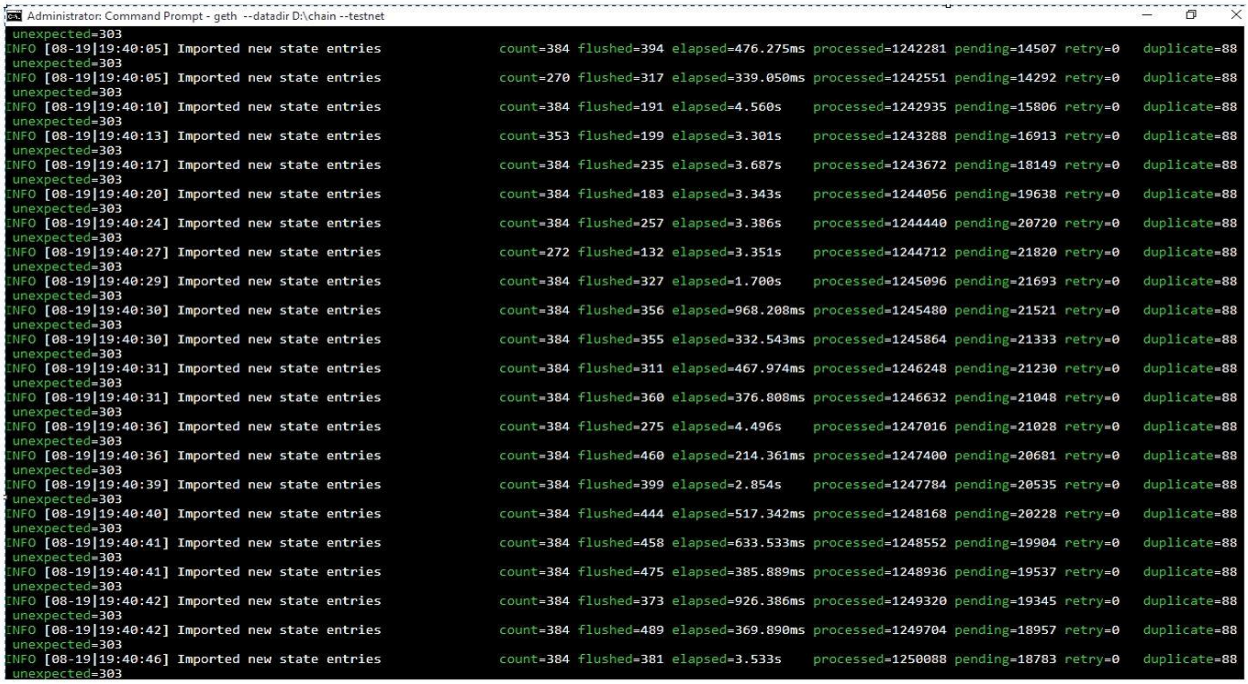

Fig. 4. Synchronization of the Ethereum network.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *25 September 2023* ***Revised:*** *12 November 2023* ***Accepted:*** *30 November 2023*

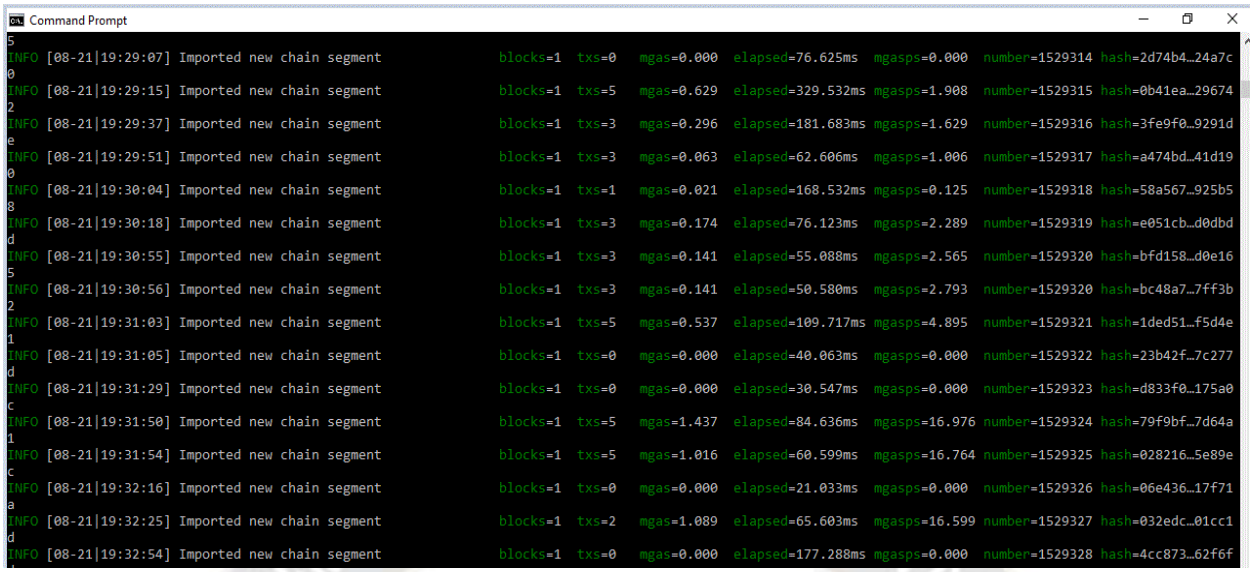

Fig. 5. Importation of new chain.

4.1 Implementation of the smart contract, Dapp, and voting wallet

The compiler transforms the bytecode to binary during compilation so that the Ethereum Virtual Machine (EVM) can read it. at order to deploy a contract to the test network, Ether must be at the default wallet address. This is because Ether provides the gas required for miners to complete the necessary computational chores. After deployment is finished, users can monitor the contract and initiate transactions on events and functions by using the Application Binary Interface (ABI) and blockchain location that are provided. The procedure entails assembling the smart contract, configuring the server node, and launching the decentralized voting program. The voting DApp module is first assembled and integrated into the workspace. As seen in Figure 6, DApp users can access a webpage that shows the ABI and system details, including the sealed and encoded voting contract. Stakeholders in the election can validate and approve the smart contract code thanks to its transparency. On the other hand, the contract becomes void if the ABI is changed without the stakeholders' consent. The contract address, ABI, and operational features are made publicly visible through the successful deployment of an online voting wallet application. As shown in Figure 8, the result is a list of voting functions released by the election authority, leading to the creation of a decentralized voting module.

Fig. 6. Publicly verifiable smart contract ABI

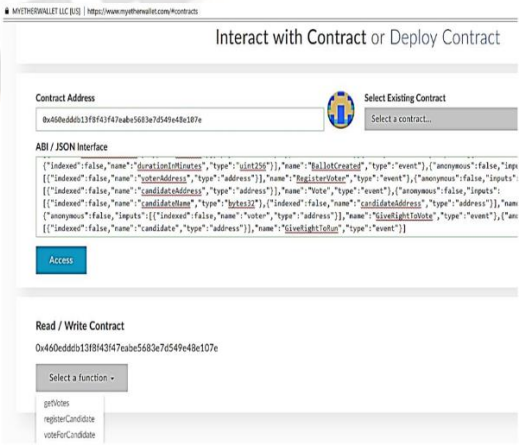

Fig.7. Ethereum voting wallet.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *25 September 2023* ***Revised:*** *12 November 2023* ***Accepted:*** *30 November 2023*

Fig. 8. Voting functions.Test EvaluationTest Evaluation

5.Test Evaluation

The implementations' test results are displayed in this section. It is evaluated for the ability to register a single voter, stop multiple votes in an easily verifiable voting process.

5.1 Test for single voter registration

It is anticipated that voters would use MyEtherWallet to open wallet accounts. They open https://www.myetherwallet.com, navigate to the contract tab, input the contract address, and copy and paste the ABI into the election authority's JSON interface. They unlock the wallet to produce a registration transaction after choosing '*registerVoter*' under the '*Access'* tab.

Figure 9 illustrates how the voting contract previews the registration's successful transaction hash. The registration process is repeated using the same wallet address to check for double registration. As shown in figure 10, the system appropriately rejects the duplicate registration attempt.

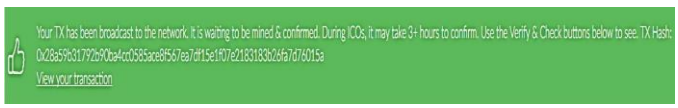

Fig. 9. Registration transaction generated; hash confirmed successful.

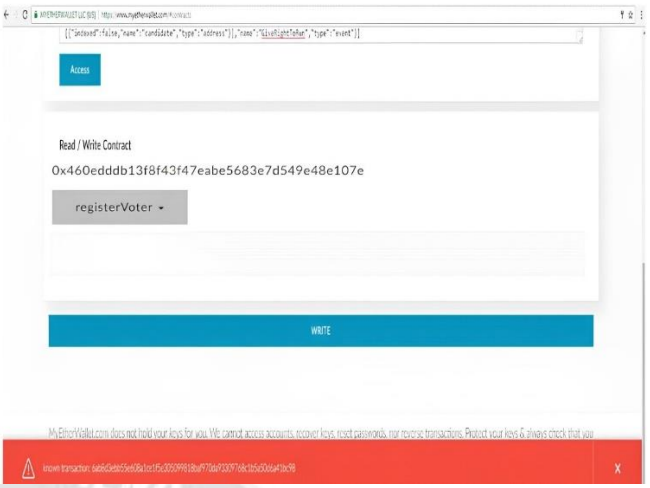

Fig.10. Rejection of double registration with system generated known transaction.

5.2 Test for single voting

Comparable procedures to those in section 5.1 are used to test against double voting. The voter selects '*voteForCandidate*' by clicking on the '*Access*' page, then uses the relevant voting mechanism. Every vote attempt is matched to a transaction hash for verification, and it is then pending blockchain network mining confirmation. But if a voter tries to vote more than once for the same candidate, the system rejects the duplicate transaction and displays an error notice that reads, '*Known transaction*' as seen in figure 11.

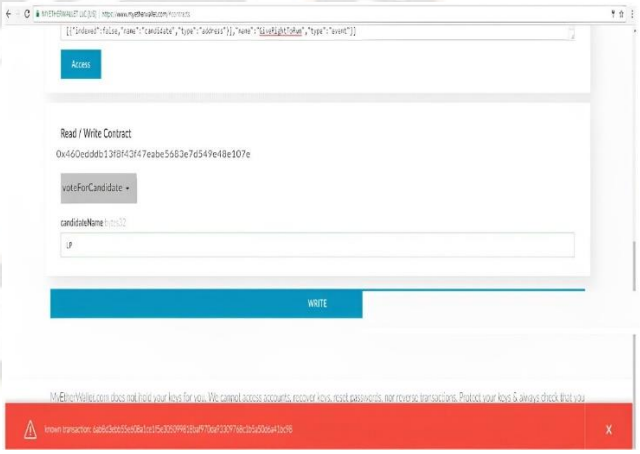

Fig.11. Rejection of double voting with system generated known transaction.

5.3 Test for election deadline and result

In addition to being evaluated for open verifiability, the election result is verified to see if it is generated automatically as each voter casts their ballot. The purpose of the test is to verify that the deadline for contract publication,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *25 September 2023* ***Revised:*** *12 November 2023* ***Accepted:*** *30 November 2023*

which calls for the prohibition of all contract transactions, will be met. Voting ends in accordance with the terms of the contract. The Dapp web application and wallet both underwent testing to verify the outcome.

5.3.1 Election result on Ethereum wallet application

The following procedures are used to verify the election deadline and results: opening the wallet, going to the '*Contract*' tab, and choosing '*Watch contract'*. Before confirming, the user then inputs the Contract name, Contract address, and JSON Interface.

After that, the election results are shown, and as figure 12 shows, the voting mechanism automatically ends when the deadline is reached. When the deadline passes and the vote is counted, the voting contract becomes unusable and cannot be accessed again.

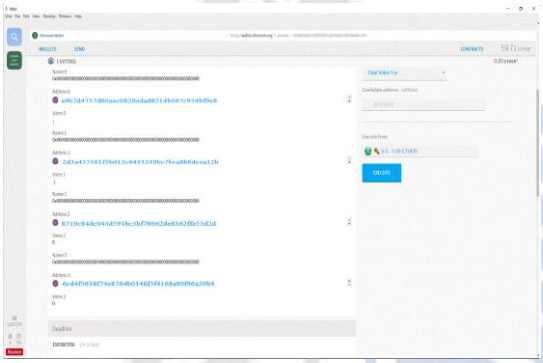

Fig.12. Election result on Ethereum wallet.

5.3.2 Election result on a decentralized web browser

Analogously, a sequence of procedures was used to test the Dapp. The voting directory was the first command line that was opened. Figure 13 illustrates what was displayed when the URL was entered into a web browser to view the result of the decentralized program.

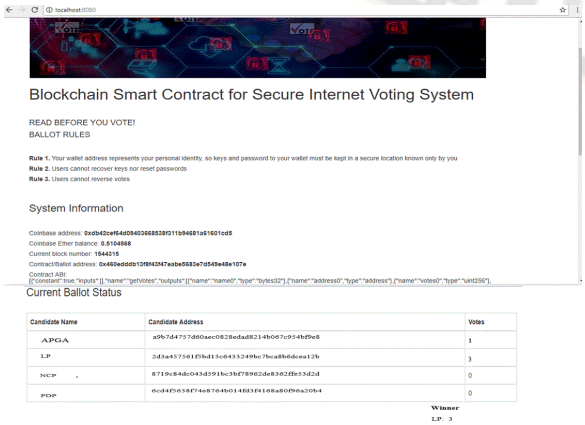

Fig.13. Decentralized election results on a web browser

6.Conclusion

To facilitate consensus among participants and validation of transactions in situations like Nigeria's council election, this research built a blockchain smart contract for online voting utilizing the Ethereum private network. The research followed the steps of creating Ethereum wallet and building a decentralized application for the election. The research also described the blockchain technology and its cryptographic and hashing protocols that ensure data integrity and immutability. The research selected Ethereum platform for its unlimited smart contracts use cases and compared them with other alternatives. The system designed a contractual code (recognized as ABI) that regulates the election process and verifies voting transactions using digital signatures and public keys. The system tested voter registration and voting transactions and found that they were secure and prevented double registration and double voting by the same wallet address.

As mentioned in [2], the test result of the blockchain technology enhances the security and reliability of internet-based voting systems, potentially revolutionizing the democratic process. The blockchain-based electronic voting protocol automates the election process and ensure secure recording and verification of votes while protecting voter identity and privacy. However, the research also recognized that blockchain is not immune to cyberattacks and suggested more cybersecurity measures, such as implementing multi-factor authentication, conducting security assessments, penetration tests, and audits of smart contracts, reducing distributed denial-of-service (DDoS) attacks, and using cold (hardware) wallet storage for future implementation. The research concluded that blockchain smart contract can improve internet voting systems and electronic democracy by enhancing their security, privacy, and anonymity.

7. Future work

The scalability and performance of blockchain smart contracts for online voting in large-scale elections require more research, according to an evaluation of the smart contract platforms in table 1. The comparison and analysis of attacks on the proposed blockchain with the conventional blockchain would therefore lead to future research paths that would improve security measures in blockchain-based electronic voting systems. Possible future research directions could include optimizing the automation processes, enhancing the architecture design, and scaling up the e-voting system for larger elections. It is noteworthy to conduct comparative studies with traditional and existing methods and exploring challenges in implementing blockchain in national elections. The research could investigate how to optimize the network bandwidth, transaction speed, resource consumption

of the blockchain system, and dependent devices. The research could also evaluate the trade-offs between security and efficiency in different consensus mechanisms and network configurations. It is important to consider what happens behind the network segments, the voting machines and vote counting software.

Other possible areas could involve the examination of the social and ethical implications of blockchain smart contract for internet voting, such as voter sensitization, digital divides, and legal frameworks. Further research is needed to examine how the technology affects the trust, transparency, and accountability of the electoral process. Where such system is present, a follow-up survey of users' perception of the system behavior for acceptance and adoption would be made. This is to provide insights into the user expectations, preferences, attitudes, and behaviors regarding blockchain smart contract for internet voting. A survey could also identify the factors that influence the user satisfaction and trust towards blockchain smart contract for internet voting. A survey could help to evaluate and innovate blockchain smart contract for internet voting systems that meet the user needs, legal requirements, and overall credibility of the election process.

References

1. E. Akbari, Q. Wu, W. Zhao, H. R. Arabnia and M. Q. Yang, From Blockchain to Internet-Based Voting (2017) International Conference on Computational Science and Computational Intelligence (CSCI). IEEE. doi: 10.1109/CSCI.2017.34
2. S. Khaleelullah, D. S. Hemanth, E. Kavitha, B. Viswadutt, and B. S. V. Teja, A Novel Blockchain based Decentralised Ballot System (2023) International Conference on Sustainable Computing and Smart Systems (ICSCSS). IEEE. doi: 10.1109/ICSCSS57650.2023.10169163
3. C. C. Z. Wei and C.C. Wen, Blockchain-Based Electronic Voting Protocol. JOIV : International Journal on Informatics Visualization (2018) 2(4-2), 336–341. doi: 10.30630/joiv.2.4-2.174
4. U. Jafar, M. Aziz, Z. Shukur, Blockchain for Electronic Voting System-Review and Open Research Challenges. Sensors (Basel). 2021, 21(17):5874. doi: 10.3390/s21175874. PMID: 34502764; PMCID: PMC8434614.
5. S. Chouhan and G. Sharma, Secure Online Voting System: Blockchain and other Approaches. ICIMMI '22: Proceedings of the 4th International Conference on Information Management & Machine Intelligence. Association for Computing Machinery (2022). doi: 10.1145/3590837.3590894
6. A. Kosba, A. Miller, E. Shi, Z. Wen and C. Papamanthou, Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts (2016) IEEE Symposium on Security and Privacy (SP), San Jose, pp. 839-858. doi: 10.1109/SP.2016.55.
7. D. Christopher, A. Vikram and B. Lee, Smart Contract Templates: foundations, design landscape and research directions. Barclays Bank PLC (2017). http://arxiv.org/abs/1608.00771
8. H. Guo and X. Yu, A survey on blockchain technology and its security. Blockchain: Research and Applications, 3(2), 100067 (2022) doi: 10.1016/j.bcra.2022.100067
9. S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 21260, 2008.
10. J. Xu, Are Blockchains immune to all malicious attacks? Financial Innovation, Springer International Publishing, Berlin Heidelberg (2016). doi:10.1186/s40854-016-0046-5.
11. Anusuya, R., Karthika Renuka, D and Ashok Kumar, L. Privacy-preserving in smart grids using Ethereum and Hyperledger blockchain. Blockchain-Based Systems for the Modern Energy Grid. Academic Press (2023). doi: 10.1016/B978-0-323-91850-3.00016-0
12. C. Catalini and S. Joshua, Some Simple Economics of the Blockchain. Rotman School of Management Working Paper No. 2874598; MIT Sloan Research Paper No. 5191-16 (2016). http://dx.doi.org/10.2139/ssrn.2874598.
13. Bitcoin. Bitcoin Developer Guide (2017). https://bitcoin.org/en/developer-guide#block-chain
14. Ethereum. Ethereum Whitepaper (2022). https://ethereum.org/en/whitepaper/
15. E. Piscini, J. Guastella, A. Rozman and T. Nassim, Blockchain: Democratized trust Distributed ledgers and the future of value. In Tech Trends (2016). Innovating in the digital era. https://www2.deloitte.com/content/dam/Deloitte/de/Documents/technology/ER_3039_TT6_Blockchain_DE.pdf
16. P. Harry, Blockchain Mining: The Key To Powering A Decentralized World." Forbes, 6 July (2018). www.forbes.com/sites/forbestechcouncil/2018/07/06/blockchain-mining-the-key-to-powering-a-decentralized-world/?sh=7c936a335a11#5626a8bd5a11%5D
17. G. Yu, B. Wu and X. Niu, Improved Blockchain Consensus Mechanism Based on PBFT Algorithm (2020) 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC). IEEE. doi: 10.1109/CTISC49998.2020.00009
18. S. Yan, Analysis on Blockchain Consensus Mechanism Based on Proof of Work and Proof of Stake (2022) International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI). IEEE. doi: 10.1109/ICDACAI57211.2022.00098

19. B. Nicolas, G. Rachid and H. Florian, Fast Byzantine Agreement. In Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC, NY, USA, pp 57–64 (2013).
20. L. Guanfeng, and V. Nitin, Capacity of Byzantine Agreement: Summary of Recent Results. In Proceedings of the (2010) ACM workshop on Wireless of the students, by the students, for the students, Chicago, Illinois, USA, pp 13-16.
21. P. Dikshit and K. Singh, Efficient weighted threshold ECDSA for securing bitcoin wallet. In ISEA Asia Security and Privacy (ISEASP), Surat, India, pp. 1-9 (2017). doi: 10.1109/ISEASP.2017.7976994
22. C. Gouvêa, L. Oliveira and J. López. Efficient software implementation of public-key cryptography on sensor networks using the MSP430X microcontroller. In Journal of Cryptographic Engineering, Springer. vol (2), pp 19-20 (2012). doi: 10.1007/s13389-012-0029-z.
23. A. Selvakumar and C. Ganadhas, The Evaluation Report of SHA-256 Crypt Analysis Hash Function. International Conference on Communication Software and Networks, Macau (2009) pp. 588-592. doi: 10.1109/ICCSN.2009.50.
24. C. Jeong and V. Kim, Implementation of efficient SHA-256 hash algorithm for secure vehicle communication using FPGA (2014) International SoC Design Conference (ISOCC), Jeju, pp. 224-225. doi: 10.1109/ISOCC.2014.7087617.
25. M. Sato and S. Matsuo, Long-Term Public Blockchain: Resilience against Compromise of Underlying Cryptography. IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Paris, France, pp. 42-49 (2017). doi: 10.1109/EuroSPW.2017.49.
26. N. Szabo, The Idea of Smart Contracts, (1997) http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_idea.html
27. Y. N. Szabo, A Formal Language for Analyzing Contracts (2022). http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/contractlanguage.html
28. C. Clack, V. Bakshi and L. Braine, Smart Contract Templates: Foundations, Design Landscape and Research Directions (2016). https://arxiv.org/pdf/1608.00771
29. K. Saini, A. Roy, P. R. Chelliah and T. Patel, Blockchain 2.O: A Smart Contract (2021) International Conference on Computational Performance Evaluation (ComPE). IEEE, 2021. doi: 10.1109/ComPE53109.2021.9752021
30. K. Christidis and M. Devetsikiotis, Blockchains and Smart Contracts for the Internet of Things. In IEEE, vol. 4, pp. 2292-2303 (2016) doi: 10.1109/ACCESS.2016.2566339.
31. R. Morgan, BitCongress - Process for Blockchain Voting & Law (2016). http://www.bitcongress.org/BitCongress_Whitepaper.pdf
32. Coindesk. Leader in Blockchain News (2017). https://www.coindesk.com/
33. B. Massimo and P. Livio, An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns. WTSC 2017. https://arxiv.org/pdf/1703.06322v1.pdf
34. Microsoft. Overview of the .NET Framework, 2017. https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview
35. NEM. NEMTechnical Reference (2018). https://nemproject.github.io/nem-docs/pages/Whitepapers/NEM_techRef.pdf
36. Ethereum. Ethereum development documentation (2022). https://ethereum.org/en/developers/docs/#top
37. Hyperledger. Hyperledger Foundation (2022). https://www.hyperledger.org/
38. Steller. Stellar Consensus Protocol (SCP). https://developers.stellar.org/docs/fundamentals-and-concepts/stellar-consensus-protocol
39. V. Martin and S. Philipp, Comparison of Ethereum, Hyperledger Fabric and Corda. FSBC Working Paper (2017). http://explore-ip.com/2017_Comparison-of-Ethereum-Hyperledger-Corda.pdf
40. M. Vukolić, The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication, 2016. In: Camenisch J., Kesdoğan D. (eds.) Open Problems in Network Security, Vol. 9591, Springer
41. G. Wood, Ethereum: A secure decentralised generalised transaction ledger EIP-150 Revision (2014). Available from: http://gavwood.com/Paper.pdf.
42. Web3j. Transactions (2022). https://docs.web3j.io/4.8.7/transactions/transactions/
43. K. Jerome, Blockchain 2.0 - From Bitcoin Transactions to Smart Contract Applications (2016). https://www.niceideas.ch/roller2/badtrash/entry/Blockchain-2-0-from-bitcoin#sec6
44. A. Blandin, G. Pieters, Y. Wu and T. Eisermann, A. Dek, S. Taylor, D. Njo, 3rd Global Cryptocurrency Benchmarking Study." The Cambridge Centre for Alternatve Finance (2020). https://www.jbs.cam.ac.uk/wp-content/uploads/2021/01/2021-ccaf-3rd-global-cryptoasset-benchmarking-study.pdf
45. Myetherwallet. Ethereum's Original Wallet (2022). https://www.myetherwallet.com/

Authors Biography

Amah Nnachi Lofty is a Lecturer in the Department of Computer Science, Federal College of Education, Kontagora,

Nigeria. He holds an MSc in Computer Security and Forensics (with distinction) from the University of Bedfordshire, United Kingdom, and obtained a Certificate in Human Factors in Systems Safety and Security from Bournemouth University, United Kingdom. He earned a BSc in Computer Science from Ebonyi State University, Nigeria, and a Postgraduate Diploma in Education from the National Open University of Nigeria. The author is a member of the British Computer Society (Chartered), the Institute of Electrical and Electronics Engineers (IEEE), and the Institution of Engineering and Technology (IET). He is pursuing a PhD at the University of Malaya, Malaysia. His research interests encompass e-learning, blockchain, electronic democracy, cybersecurity, and digital forensics.

Ali Mansour is a Senior Lecturer in Computing and Information Systems at the University of Bedfordshire. He has a BSc (Honours) in Computer Studies from Sheffield City Polytechnic (now Sheffield Hallam University) in 1987, and a PhD in Computer Science from the University of Sheffield in 1990. He is a Fellow of the British Computer Society (FBCS), an Engineering Council Chartered Engineer (CEng), a BCS Chartered IT Professional (CITP), and a Fellow of the Higher Education Academy (FHEA). His research areas are in cybersecurity, digital forensics, networking, e-learning, and medical informatics.