Journal of Theoretical and Applied Information Technology
30th September 2022. Vol.100. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPLEMENTING A SECURED OFFLINE BLOCKCHAIN BASED ELECTRONIC VOTING SYSTEM

APEH JONATHAN APEH[1], CHARLES K. AYO[2], AYODELE ADEBIYI[3]

[1] Covenant University, Ota, Ogun State, Nigeria
[2] Trinity University, Yaba, Lagos State, Nigeria
[3] Landmark University, Omu-Aran, Kwara State, Nigeria
Email: [1]apeh.jonathan@gmail.com

ABSTRACT

Internet penetration is a critical factor in determining the adoption of an internet-based electronic voting system. In Nigeria for instance, the electoral bill which is to provide the legal backing for the transmission of voting data from polling units across the federation had witnessed huge setbacks hinged on lack of internet infrastructure in Nigeria's remote places. Arguments have been made for and against the possibility of disenfrenching citizens if electronic transmission was allowed in the electoral laws. In this article, we propose an electronic voting system model that allows the secured continuation of voting exercise in offline mode and the transmission of cast votes upon the restoration of internet connectivity using the infrastructures permissible with the blockchain technology. The model is adaptable in any national election, specifically, Nigeria's national elections.

Keywords: *Blockchain, Election, Electronic Voting System, Internet penetration, Suffrage*

1. INTRODUCTION

Electronic voting system occupies a special position in birthing a truly democratic leadership in any nation. Whether they are internet-based or not, the result they produce, determine who occupies the public offices where the quality of life of a people can be decided, where their dreams and aspirations can be determined. It also serves as a veritable tool to determine the economic prosperity of a nation. This is why the e-voting system is considered a critical national infrastructure.

However, most deployed e-voting systems have been found to be vulnerable to varied degrees. These flaws have the ability to sabotage elections, compromise election outcomes, and even risk the confidentiality of voters' votes. E-voting systems are vulnerable to a wider range of threats than traditional voting systems. This includes everything from insider attacks by system administrators, cybercriminals, and hacktivists who disrupt the voting process as a form of political protest to sophisticated hackers who engage in offensive cyberwarfare against the e-voting system [1]. Denial of service assaults, vote spoofing, and voter phishing are examples of other types of attacks [2]. It is this understanding that is driving the different research going on to improve the security of the e-voting system.

The application of the blockchain technology to the e-voting system is one of the recent research in this regard. Blockchain technology is a technology powering distributed database of blocks of transactions. The technology is built on cryptography and distributed computing. On the blockchain network, each transaction is verified before they are added to the block, each block is validated before they are added to the chain of blocks by leveraging cryptographic infrastructures that enhances the security of the transactions. With cryptography, each transaction hash is added to the parent hash of all preceding transactions hash. This is the basis for the Merkle tree structure of the blockchain. With this security structure of the blockchain, it is extremely difficult to alter a transaction by seeking to manipulate it.

An electronic voting system built on blockchain technology is internet-based and leverages the security features available through the blockchain technology. These features include smart contract, decentralization, immutability, transparency, auditability.

Smart contract is "a computerized transaction protocol that executes the terms of a contract." [3]. The general objectives of smart contract design are to satisfy common contractual conditions such as payment agreements, liens, confidentiality, and even enforcement; minimize exceptions both malicious and accidental, and to minimize the need for trusted intermediaries [4]. Smart contract simply implement the business logic deployed on the blockchain. Decentralization on the other hand is that feature of the blockchain technology that allows it to function without a central controlling authority [5]. This means every computer peer on the network contributes equally to the decision making process on the network.

Immutability makes it impossible to change the existing or committed transactions on a blockchain network without being detected. This is possible because the blockchain network is built on cryptography to ensure the security and integrity of data and transactions. Transparency is another important feature of blockchain technology as it allows transactions on a blockchain network to be visible to only authorized participants. This means the technology enhances auditability and trust while reducing the cost of fraud and audits. Auditability feature on the blockchain promotes transparency and trust as all transactions are available for scrutiny and investigation.

2. LITERATURE REVIEW

2.1. Internet Penetration in Nigeria

Internet penetration is the percentage of the population of a country or region with access to the internet. It is a variable that is very critical to economic development of a people. Internet penetration in a place can either increase or decrease. An increase in Internet penetration can engender momentous growth in vital sectors of the economy such as education, healthcare, energy, and governance.

Statista [6] represented the internet penetration rate in Nigeria. According to the report, as of the year 2021, 51.44% of Nigeria's population has access to the internet. This shows that over 50% of the Nigerian population have access to the internet. However, issues such as epileptic power supply, inadequate and poor internet infrastructure are militating against the steadiness and expansion of internet infrastructure around Nigeria.

This could be because of poverty or lack of the necessary infrastructure upon which internet services ride. Acknowledging the threats that the internet connectivity deficit could cause a national event like voting which has a lot of significance to democracy, security of lives and properties, peace and tranquility of the nation, the proposed design embeds the possibility to vote unhindered by lack of internet access [7]

2.2. Nigeria's Internet Infrastructure and current Voting System

Internet technology rely on a range of devices, systems and other technologies to be delivered. These includes fiber networks, backbone, Internet Service Providers, routers, gateways, power supply, etc. These are generally termed internet infrastructures.

The availability or proximity of internet infrastructure to a place can determine the accessibility and availability of the internet to the population in that location. In Nigeria for instance, there is a broadband service delivery which is made up three layers: the first mile, national backbone, the middle mile network, and last mile.

The first mile, otherwise known as the international links, covers over 1000 km, it helps to connect the service centers to the worldwide internet using satellite, terrestrial fiber, subsea cable and the worldwide web(www). Whereas the national backbone tier helps to connect bigger cities together using fiber. It is used to cover distance of 100 to 1000 km. The middle mile network is for distribution. It is used for bringing internet to a point in a community for broader distribution using fiber, microwave and satellite link. It has a coverage of 10 to 100 km. The last but not the least is the last mile or access tier. This helps to distribute internet from the middle mile in the community to individual home and businesses using fiber, DSL, COAX, wireless, Wi-Fi, LTE, TV while space, balloons, and drones. It has 1 to 5 km coverage.

In spite the elaborate arrangement for internet accessibility and availability, the last mile tier which brings internet closest to the people and businesses has not been adequately deployed. It is same tier the proposed electronic voting system rely on at the polling units for internet access [8-11]

2.3. Blockchain overview

Blockchain technology was first introduced by someone with the pseudonym, Satoshi Nakamoto [12] in a paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" [13-16]. The paper establishes the mathematical foundation for the bitcoin cryptocurrency.

Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system. A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across a network of computers called nodes [17].

It uses distributed computing and cryptography to securely host applications, store data, and easily transferred valuable digital instruments like money, artworks, cast votes, etc. [18].

It uses consensus (i.e., community validation) to keep transactions synchronized and the ledger distributed across multiple nodes in a way that is decentralized, transparent, immutable, auditable, persistent and secured [19-21]. It removes trust concerns amongst users.

A blockchain is a special type of distributed database. Whilst a distributed network works well when all entities trust each other and do not want to keep duplicate records of the same data, Blockchain however, comes into play when the entities cannot trust each other, that is, there is no single entity in control and we need a database that is distributed and decentralized. One of the building blocks of the blockchain is the transaction block. A block contains transactional data and are synonymous to ledger page. Each block contains a timestamp and a hashed link to a previous block, creating a chain of blocks. By design, blockchains transaction data are difficult to modify; once a new entry has been recorded, the data in that block cannot be altered [4]. Furthermore, older blocks cannot be altered without breaking the chain to every block that is recorded subsequently. If an attacker would attempt to modify a block, he would have to change all blocks that happened afterwards to the most recent block. This attack is very difficult to achieve [23]

Di pierro [24] posits that blockchain technology is not just at the foundation of all cryptocurrencies but also has wide application in the more traditional financial sector. It is also leading groundbreaking applications in domains such as smart contracts, the Internet of Things, Public Services, Digital Content Management, Electronic Voting, etc. Blockchain technology basically and fundamentally solves the problem of establishing trust among peers in a distributed system, hence businesses that require high reliability and honesty will find it attractive to gain customers' trust [21].

Blockchain technology utilizes cryptographic hash functions and digital signatures to maintain the integrity of all blockchains [25-26].

2.4. Merkel Tree

Merkel tree also known as binary hash tree is a data structure that has interesting applications in different computer science domains like blockchain, InterPlanetary File System (IPFS), etc.

A merkle tree is a technique for hashing a larger chunk of data (e.g., transactions) into a single hash. A block for instance holds batches of valid transactions that are hashed and encoded into a merkle tree. Merkle trees are a essential component in the blockchain technology. With it, each block in the blockchain contains a summary of all the transactions in the block. This explains the benefit of merkle tree in which it allows small and simple smart phones, laptop and even internet of things devices to run a blockchain instead of only powerful computers. In theory, it is possible to create a huge block that directly contains every transaction in the header. The downside to this is that it poses a large scalability challenges that arguably puts the use of blockchains out of reach of all but the most powerful computers.

Beyond the computational efficiency (i.e little storage requirement, enables verification without sending all, heavy application in SPV i.e. light clients) of the merkle tree, the data structure is temper-proof and helps in protecting the integrity of data. computationally efficient. In bitcoin and other cryptocurrencies for instance, Merkle trees serve to encode blockchain data more efficiently and securely.

Figure 1.0 represents the basic structure of the merkle tree. Every transaction added to a block of transactions is hashed. Let's say there are eight transactions in a block, each of the transactions, t1..8 produces a hash, h1..8. The eight transaction hashes pair up to form four hashes. On the second level, the four hashes pair up again to create two more hashes. These two new hashes again pair up to create the root hash. This structure is referred to as merkle tree. The root hash of each block, is hashed with the parent hash of existing blocks to form the whole blockchain merkle tree.

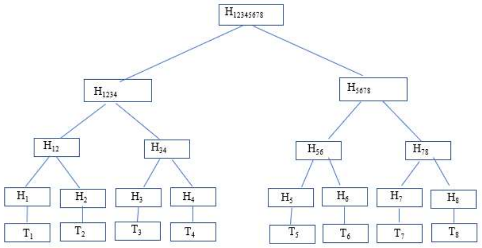


Figure 1.0: The Merkle Tree Structure[27]

2.5. Consensus Model

Consensus means agreement. Consensus model are set of rules that governs the way transactions are accepted into a block and blocks unto the blockchain network. These rules could include conditions to accept the transaction, the difficulty of the complex mathematical calculation, incentives or reward for getting the calculations correctly, etc.

The core functions of the consensus model on the blockchain network include ensuring and maintaining decentralized governance, quorum structure, authentication, integrity, non-repudiation, byzantine fault tolerance, and performance [28].

Sankar, Sindhu and Sethumadhavan [28] posit that consensus algorithms are the core of the blockchain technology. The essence of a consensus algorithm in a blockchain is to ensure the consistency of transactions across the whole P2P network. Consensus algorithms work based on the Byzantine Generals (BG) Problem. According to [21], "In BG problem, a group of generals who command a portion of Byzantine army circle the city. Some generals prefer to attack while other generals prefer to retreat. However, the attack would fail if only part of the generals attacks the city. Thus, they have to reach an agreement to attack or retreat."

For better understanding of the way the consensus model works to achieve consensus, the first understanding would be how transaction recorded in a block is decided by the miner that generated the block. Once a miner succeeds in generating a block, it receives the cumulative amount of fees from transaction. It also receives the block reward (which is known as a special transaction that always appears first in every block). This is credited to their own wallet. The block reward transaction explains how new bitcoins are created in the system. Basically, to generate a new block and claim a reward, miners compete to solve a complex mathematical puzzle and provide nonce (an answer that is unique to each block). A block cannot be submitted to the blockchain without the correct answer. Having the correct answer is what is termed as the Proof-of-Work. This concept helps to creates a distributed trustless consensus and resolves the double-spend problem [6]. Proof-of-Work is an expensive and time-consuming but easy to verify piece of data. To find the correct answer, a miner must hash to a value less than the current target. The difficulty of this work is often changed or increased to limit the hash rate to one block every ten minutes. The networks hash rate is what determines if the difficulty should increase. The more miners join the Bitcoin network, the higher the network hash rate is. Because successful generation has such a low chance, it's impossible to forecast which miner in the network will be able to generate the next block. This results in a Bitcoin distribution that is arguably fair. The blockchain encapsulates a vast amount of labor because each block contains the hash of the previous block. Changing a block by creating a new block that references the same predecessor necessitates the regeneration of all successors as well as the redoing of the enormous labor they represent. This is where the blockchain's integrity is safeguarded. Propagating a newly created block over the blockchain network takes time for a miner. Another miner may be able to solve the riddle and locate the correct solution, resulting in the generation of a new block. When this happens, each miner will start mining on the block they received first, resulting in a blockchain fork. Since the block reward and transaction fees are only recoverable if the block is part of the longest chain, honest miners only add onto it. With each mined block, the likelihood of the other miners on the various chains continuing to solve blocks simultaneously decreases, until one chain eventually overtakes the other and that chain is abandoned. Miners would not risk solving a block that was not part of the longest chain. Miners will be motivated to work toward a single blockchain version as a result of this.

Some consensus algorithms in use today are Proof of Work (PoW), Proof of Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), Delegated proof of stake (DPoS), Riddle, and Tendermint. They all compare differently when considered against properties like node identity management, energy saving, tolerated power of adversary and examples of blockchain applications using them.

i. **Proof of Work (PoW):** This is the type of consensus strategy used in blockchain networks like the bitcoin network. In PoW, the hash value of the block header is calculated by each node of the network. The block header which contains a nonce and miners changes the nonce frequently to get different hash values[28]. To achieve consensus, it is required that the calculated value must be equal to or smaller than a certain given value. In practical terms, in a PoW, when one node reaches this target value, it would broadcast the block to other nodes and all other nodes must mutually confirm the correctness of the hash value. If the block is validated, other miners would append this new block to their own blockchains. The nodes that do the calculation of the hash values are called miners, whereas the PoW procedure is called mining in Bitcoin.

ii. **Proof of Stake (PoS):** The Proof of Stake is a consensus protocol base on stakes. Blockchain networks with PoS, the miners will have to prove ownership of the amount of currency they have. This is because there is a general belief that people with more money, are less likely to attack the network. Even though the PoS is an energy-saving protocol compared to the PoW, it has its downside. The PoS in effect is base on the account balance and it is quite unfair because it allows the richest account owner to dominate the network.

iii. **Practical Byzantine Fault Tolerance (PBFT):** Fault Tolerance is the ability of a system to continue to function (even if with a reduced rate or efficiency) when it develops a fault. Distributed systems that are meant to be sustainable must be fault-tolerant (i.e. able to tolerate faults, resilient to failure). Having redundancy in place is the most intuitive way to achieve fault tolerance in a system. Redundancy allows for multiple nodes having the same compositions and states operating together. With redundancy, when there is a failure in a single node, the system can still go on operating smoothly. The PBFT is a characteristic of distributed systems. It allows distributed systems to continue to run without been completely brought to a halt. There are 3 phrases of the PBFT protocol: pre-prepared, prepared, and commit. In each phase, a node can only enter the next phase if it has received votes from over 2/3 of all nodes. The PBFT works base on the Byzantine General Problem where a group of Byzantine Generals plans to besiege a city and take it down. Each general has the choice to attack or retreat; communication between them is through a messenger. To attack or retreat, the generals will have to reach a consensus to make a decision. This is what happens in a blockchain network where the PBFT is used as a consensus algorithm. The miners on the network have to agree on a transaction that is valid or not for it to be admissible on the network.

Hyperledger Fabric utilizes the PBFT as its consensus algorithm since PBFT has the capacity to handle up to 1/3 malicious byzantine replicas.

iv. **Delegated proof of stake (DPoS):** DPOS is another consensus algorithm that uses transaction representation to secure the blockchain network. It was designed to implement a technology-based democracy using the voting and election process to protect a blockchain network from centralization and malicious usage. Unlike the PoS which is a direct democratic algorithm, the DPoS is representative democratic. This implies that stakeholders in a DPoS elect their delegates to generate and validate transaction blocks. With fewer nodes needed to validate blocks, there is faster transaction confirmation. The DPoS is used to power BitShares which was the first implementation of the DPoS

v. **Ripple** is another consensus algorithm. It makes use of collectively-trusted subnetworks within the larger network. Nodes in a ripple based network are divided into two types: server and client; the servers are responsible for participating in the consensus process while the clients are in charge of funds transfer. Every server has a Unique Node List (UNL). The servers make use of the UNL when determining whether to put a transaction on a ledger. They rely on the UNL to query the nodes in them. If from the query result from the UNL, the received agreements reached 80%, the transaction is added to the blockchain. For a node, so long as the percentage of faulty nodes in the UNL is less than 20%, the ledger remains correct.

3. METHODOLOGY

Here we present the processes and methods followed to design and implement the proposed offline blockchain model for Electronic Voting Systems. In this work, the organization and conduct of elections in Nigeria is considered hence, Nigeria's election management body, the Independent National Electoral Commission (INEC) is used as a case study.

Figure 2 represents the network view of the proposed model. It is made up of the federal blockchain and the network of polling units' nodes across the states of the federation including the federal capital territory.

While the network of polling unit nodes is made up of all the Polling Units (PU) in each state and the State Miner(node), the federal blockchain constitutes of the 36 state miners (each representing the state in Nigeria) and the Federal Capital Territory (FCT), Abuja.

The state miners are full nodes, the PU nodes are lightweight. A full node has a complete copy of the distributed ledger, validates new vote blocks, and verify transactions (in this case, the cast votes). The distributed ledger contains information about all registered voters and cast votes in the country.

Due to latency consideration, the PU nodes (i.e. lightweight nodes), do not store a full copy of the blockchain. They only keep a copy of the block header (i.e., metadata) which is updated from time to time to stay upto date with the main blockchain and to verify transactions.

With the Web3 API, a copy of the blockchain with only eligible voters in every state is made available at the respective state PUs. Communications between state miners and the PUs are digitally signed. Every PU sends a payload that is signed with its public key. With this, voters can be verified during voting before they cast their votes. The cast votes are sent as a payload which is digitally signed with each PU public key.

If peradventure, the PU loses connectivity to the blockchain, voting could be done offline, while verification gets done online later once connectivity is reestablished. For this to happen, the cast votes are captured in a MongoDB (a noSQL database) as signed transactions. The signature on the transactions in the database provide another security layer to secure cast votes against tempering in the database. The cached signed transactions are re-sync every 1 minute. This helps to check for internet connectivity.

The implication of the model's offline voting capability is that in a double voting situation, a wrong actors could vote offline but when the PU connects back, such votes would be invalidated because they had voted.

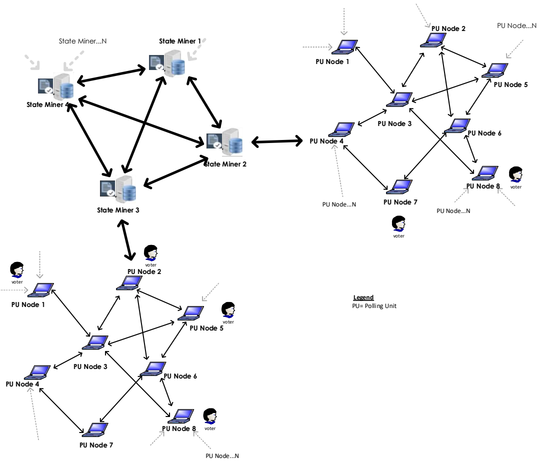


Figure 2: Network View Of The Blockchain-Enabled Model For Electronic Voting System [28]

3.1. Model Setup

The following steps were taken to set up the proposed secured offline blockchain-based electronic voting system for national election: First node set up on the private blockchain, network creation, smart contract deployment, polling units (PU) module deployment.

3.1.1. First Node Setup

A blockchain is a collection of computers with duplicate records of data, all communicating and synchronizing over peer-to-peer communication channels. These computers are referred to as nodes. To set up our private blockchain, we started with the first node. Our proposed model's proof of concept (PoC) setup has three full nodes which are called state miners. Their names are presented below, starting with the first node: eth-node1.com, eth-node2.com, eth-node3.com

These are Ubuntu EC2 nodes set up in the Amazon cloud i.e. they have installed the Ubuntu Operating System. On top of the Linux-based (Ubuntu) webserver is installed the Ethereum "Geth" application. To connect to each of these nodes, the following command is issued using a Git bash tool:

```
$ ssh -i key-pair.pem user@eth-node1.com
```

The ssh stands for secure shell. It is a secure protocol that supports the encrypted transfer of data between two computers. The key-pair.pem file is the identity file with the public key of the eth-node1.com node for a secured connection between the source machine and the blockchain node. Ubuntu is the user with which the blockchain nodes are being accessed.

Geth is installed to set up a custom/private Ethereum Node. To install "Geth" the command below is run from the node terminal: # sudo add-apt-repository -y ppa:ethereum/ethereum

The above command adds the Ethereum repository to the operating system source list. The command following command set updates the repositories and installs Ethereum alongside its command line (CLI) environment "Geth".

```
# sudo apt-get update
# sudo apt-get -y install Ethereum
```

Usually, the block holds data and transaction records, each block has a limited size and it's chained to a pre-existing block hence producing a chain of blocks. In our case, there is no pre-existing block so it has to create what is known as the genesis block.

3.1.2. Creating the Private Network

To create a network, first, a new node has to be created, then a second node to communicate and synchronize with the first node. Another alternative is replicating the above process with the same genesis file and chain ID. Once the node has been created, run *geth --datadir "./db" --networkid 1947 console* to initialize the node and access its console.

3.1.3. Deploying Smart Contract

A smart contract is required to be in place to have a decentralized application. It is a self-executing contract. It is deployed to autonomously enforce an agreement between multiple parties, in this case, an election. Because of the nature of the blockchain, a smart contract like blockchain transactions cannot be modified, manipulated, or revoked once it has been deployed is required.

Remix and Ethfiddle are tools required to build and deploy a smart contract. While EthFiddle is a browser-based solidity Integrated Development Environment(IDE) that allows users to write and test their solidity codes without deploying to a blockchain, R*emix*, also a browser-based IDE, generates the bytes codes used to deploy the smart contract and ABI data which used to interact with the smart contract over the RPC protocol.

When the byte code of the contract is generated on *Remix* a transaction is created and broadcasted to the network and in response, a transaction hash is generated, this hash acts as a transaction reference that can be used to get the transaction details. The contract byte data is first read from a file and is used to create a transaction object. The Chain ID is the same as the network's Chain ID. The transaction is then signed using a private key and sent to the blockchain network.

Upon propagation, a view into the transaction using the transaction hash generated above would reveal the new contract's address. This address is copied into the *"app-config.json"* file of the NodeJS application.

3.1.4. Deploying Polling Unit Module

Until this point, all we have done is setting up and deploying the state miners on the blockchain network. This is important as the PU module will need the state miner blockchain network for connectivity, voter's verification, cast votes validation and adding them to the blockchain. As represented in section 3.1, the PU is the module of our proposed model that is deployed at various polling units across the country. This module is installed and configured on all PU computing devices (Hewlett Packard laptops used for our proposed system) that will be used for the voting on the day of the election. To demonstrate the proof of concept, ten laptops used by service integrators (SIs) were used. These SIs are at different locations, tech-savvy, and well-educated.

The proposed design as represented in section 3.1, is meant to ensure offline voting in the absence of internet connectivity. To demonstrate how this objective is achieved, we need to showcase how the votes are cast at different PUs even when there is no internet connectivity. But to do that, we must first demonstrate how to setup the PU devices from which voting would be done.

Setting up the PU Device

To setup the PU device, the SIs download the PU module from the OneDrive repository, extract and navigate to the folder where the packaged PU is stored as shown above and double click on the "run.bat" file. This installs and connects the PU device to the state miner that it has been configured within the config.json file to connect to as shown in the Figure 3.1.

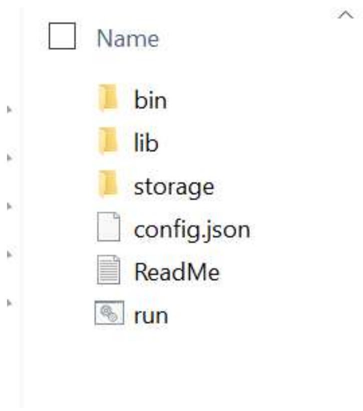


Figure 3.1: PU Module Directory Structure

Once the setup is completed, the eVoter web module interface is loaded on port 5000 as shown in Figure 3.9.

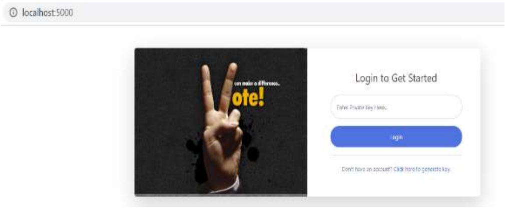

localhost:5000
can make a difference...
Vote!
Login to Get Started
Enter Private Key Here...
login
Don't have an account? Click here to generate key

Figure 3.1: Evoter Web Interface

The config.json does not just specifies the address of the state miner that each PU should connect to, it also specifies other critical parameters including: the private key and address of the state miner the PU is connecting to, contract details; the MongoDB configuration details; the provider parameter defines the state miner which the PU is securely communicating with. This variable also determines the state in which the PU is. More information about the content of the config.json file is presented in the code snippet below:

```
{
        "debug_mode": false,
        "base_url":
"http://localhost:5000/",
        "timezone": "Europe/Paris",

        "credentials": {

                "private_key":
                "0xd09e25fe665a85b
                a6477b027cb4282070
                e8989fd52ae9bfb8f97
                230329f16506",

                "contract":
"0x83Ed9591534a6c6aE32e061AF7948
0DdE2CAdFED",

                "address":
"0xeacce0d0850495888184f050bb1d36
b006f9aa81"

        },

        "provider": "http://eth-
node1.com:8545",

        "ropsten_provider":
"https://ropsten.infura.io/v3/3bee5bda21
4d46e3814c4bca369bb11e",

        "mongo.config": {

                "database" :
"e_voters",

                "host": "127.0.0.1",

                "port": 27017,

                "ttl": 1209600

        },

        "http.request": {

                "secure": false,

                "host": "127.0.0.1",

                "path": "/api",

                "port": 80

        }

    }
```

To demonstrate that voting can be done without internet connectivity, the proposed design makes use of MongoDB on the PUs to cache voting transactions. This is necessary to provide a temporary memory for voting data before they are transferred to the main blockchain. This feature is built into the design not just to improve the model's latency but to ensure that the system is not brought to its knee in the absence of internet connectivity. Cached transactions in the MongoDB are represented in Figure 3 below.

The transaction cache contains signed voting transactions made up of ObjectId, key, object and timestamp. If any of these information is tempered whilst in the cache due to internet connectivity concern, the voting transaction will not be accepted on the blockchain as the key would have changed.

Below code snippet, generates signed transactions for cast votes when there is internet failure or when system is offline:

```
Function (err, transactionObject){
console.log ({ error: 500, message: err.toString() });
res.send({ error: 500, message: "No connection at the
moment, your request would be pushed once
connection is re-established. Thank you." });
let asyncStorage = require('../model/async-storage');
let database = new asyncStorage.database();
let key =
req.params.address+'.'+generateRandomKeyPrefix();
database.insertOne('TransactionCache', { key, object:
transactionObject.rawSignedTransaction, timestamp:
Date.now().toString() });
});
```

After the cast vote transaction is signed, it is temporarily logged or cached in the MongoDB as represented in the code snippet below:

```
const EthereumTx = require('ethereumjs-
tx').Transaction;
privateKey = Buffer.from(privateKey.replace('0x', ''),
'hex');
let ethereumTx = new EthereumTx(transactionObject)
```

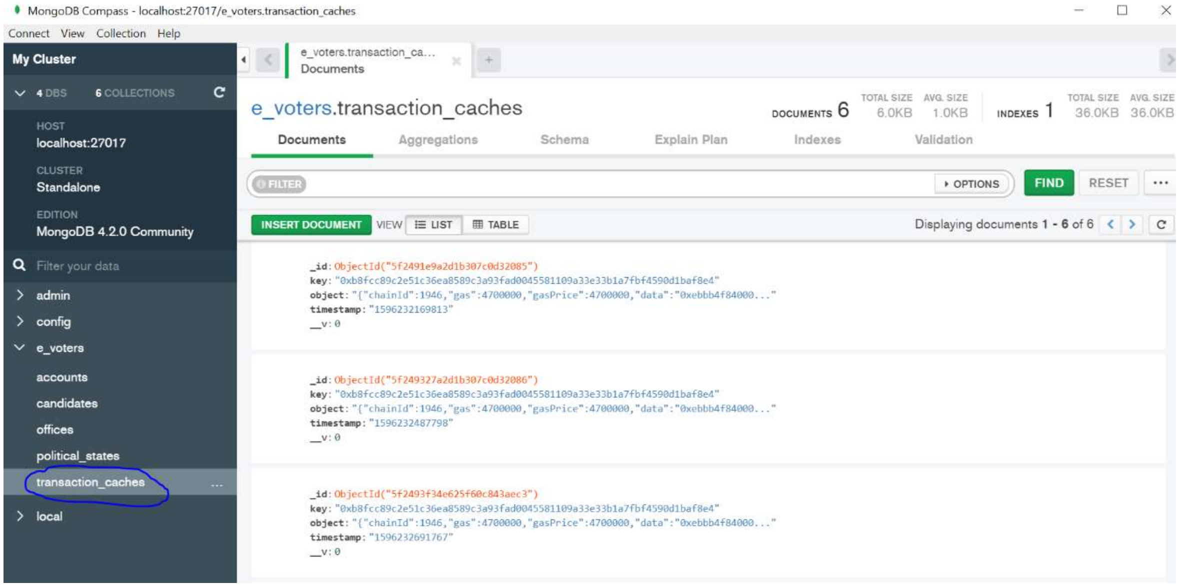
```
transactionObject.rawTransaction =
'0x'+ethereumTx.serialize().toString('hex');
ethereumTx.sign(privateKey);
transactionObject.rawSignedTransaction =
'0x'+ethereumTx.serialize().toString('hex');

errorCallback(err, transactionObject);
console.log(err);});
```

Cached transactions re-sync is triggered every 1 minute in anticipation that internet connectivity has been restored. The code snippet doing the re-sync is shown below:

```
cron.schedule('1 * * * *', function(){
  // update every 1 minutes
console.log('here here');
let contractManager = require('./controller/contract-
manager');
contractManager.retryCachedTransactions();
```

This session handles the retry:

```
let retryCachedTransactions = function(){
let asyncStorage = require('../model/async-storage');
let database = new asyncStorage.database();
database.select('TransactionCache', {}, function
(transactions) {
      console.log("Attempting transaction update
push");
      if(transactions && transactions.length > 0){
      let promises = [];
      const Promise = require('bluebird');
      for(let i=0; i<transactions.length; i++){
      promises.push(new Promise(function(resolve){
      const transaction = transactions[i];
      _sendSignTransaction(transaction.object, (err,
txHash) => {
if(!err) database.delete('TransactionCache', {
timestamp: transaction.timestamp, key: transaction.key
});
console.log('err: ', err, 'txHash', txHash);
resolve();
});  }));
}
Promise.all(promises).then(function(){
      console.log("Transaction push attempt
completed");
      });
}
      });
};
```



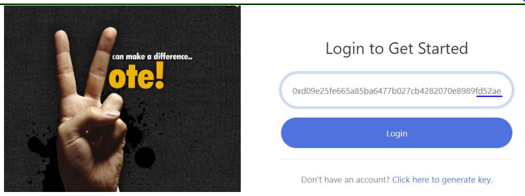
```
_id: ObjectId("5f2491e9a2d1b307c0d32085")
key: "0xb8fcc89c2e51c36ea8589c3a93fad0045581109a33e33b1a7fbf4590d1baf8e4"
object: "{"chainId":1946,"gas":4700000,"gasPrice":4700000,"data":"0xebbb4f84000..."
timestamp: "1596232169813"
__v: 0

_id: ObjectId("5f249327a2d1b307c0d32086")
key: "0xb8fcc89c2e51c36ea8589c3a93fad0045581109a33e33b1a7fbf4590d1baf8e4"
object: "{"chainId":1946,"gas":4700000,"gasPrice":4700000,"data":"0xebbb4f84000..."
timestamp: "1596232487798"
__v: 0

_id: ObjectId("5f2493f34e625f60c843aec3")
key: "0xb8fcc89c2e51c36ea8589c3a93fad0045581109a33e33b1a7fbf4590d1baf8e4"
object: "{"chainId":1946,"gas":4700000,"gasPrice":4700000,"data":"0xebbb4f84000..."
timestamp: "1596232691767"
__v: 0
```

Figure 3: Mongodb Transactions Cache

4. RESULTS AND DISCUSSION

Base on the aim of this research work which is to advance a secured offline blockchain-enabled electronic voting model, we present below outcomes from the experimentation conducted and represented in section 3.

A voter login into the mode e-voting platform with their private key: *0xd09e25fe665a85ba6477b027cb4282070e8989fd52ae9bfb8f97230329f16506* as show in Figure 3 below.



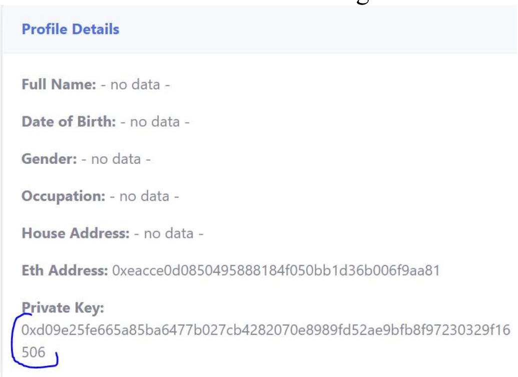
Login to Get Started

0xd09e25fe665a85ba6477b027cb4282070e8989fd52ae

Don't have an account? Click here to generate key.

Figure 3: eVoter login page

After successful login, the user's profile is shown with user's details as in Figure 4



Profile Details

Full Name: - no data -

Date of Birth: - no data -

Gender: - no data -

Occupation: - no data -

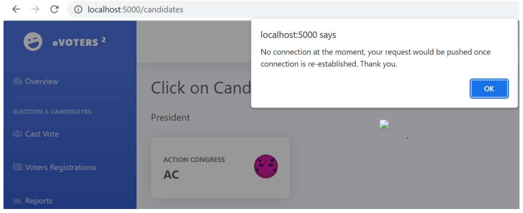
House Address: - no data -

Eth Address: 0xeacce0d0850495888184f050bb1d36b006f9aa81

Private Key: 0xd09e25fe665a85ba6477b027cb4282070e8989fd52ae9bfb8f97230329f16506

Figure 4: Voter's Profile

Without internet connection, the user tries to vote for the AC, a political party as shown the Figure 5 as shown below and a message pops up showing there is no connection at the moment.

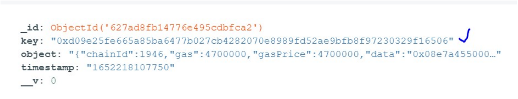


localhost:5000 says

No connection at the moment, your request would be pushed once connection is re-established. Thank you.

Figure 5: Voting Without Internet Connectivity

Even though there is no connection at this time, the voter's vote which is encapsulated in the object's variable is signed (i.e. encrypted) and cached in the MongoDB as represented Figure 6 below. The key variable on the MongoDB is consistent with the private key used in login below. Any attempt to temper with the encrypted object's data invalidates the caste vote when the PU reconnects back to blockchain for synchronization.



```
_id: ObjectId('627ad8fb14776e495cdbfca2')
key: "0xd09e25fe665a85ba6477b027cb4282070e8989fd52ae9bfb8f97230329f16506"
object: "{"chainId":1946,"gas":4700000,"gasPrice":4700000,"data":"0x08e7a455000…"
timestamp: "1652218107750"
__v: 0
```

Figure 6: Voting Data Cache

5. CONCLUSION

In this paper, a secured offline blockchain-based electronic voting system model was presented. The design showcased how a secured offline blockchain model for electronic voting system is evolved.

The model's network view which is made up of two logical blockchain networks was represented- the blockchain network and the network of polling units. We demonstrated how the model is setup right from the first node to the deployment of the smart contract and the polling units.

The model demostrated how voting goes on securely in the absence of internet connection using the MongoDB cache that supports in capturing signed voting data. The cache has the capacity to store the signed data without compromising their integrity.

The model has been demonstrated to be a panacea for poor internet connectivity and penetration challenges that hamper the deployment of electronic voting system in communities in Nigeria and around the world.

6. FUTURE WORK AND LIMITATION

Future work on this piece of research would be in the direction of deploying the polling units module on mobile devices with capacity for voting to continue without data connections. This is the currently limitation with this work.

REFERENCES

[1] Hao, F. & Ryan, P. Y. A. (2016). Practical Attacks on Real-world E-voting. In Hao, .F & Ryan, P. Y.A (Eds), Real-World Electronic Voting: Design, Analysis and Deployment (pp.145-196). US: Auerbach Publications

[2] http://www.joiv.org/index.php/joiv/article/view/174

[3] Szabo, N. (1994). Smart Contracts. Retrieved 17 July, 2020, from https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[4] Darra, L.H. (2017). Legally Speaking: Smart Contracts, Archival Bonds, and Linked Data

in the Blockchain. IEEE Computer Society, 1-4.

[5] Xu, D., Shi, W., Zhai, W., and Tian, Z. (2021). Multi-Candidate Voting Model Based on Blockchain. *IEEE/CAA Journal of Automatica Sinica*, 8(12), 1891-1900, December 2021. doi: 10.1109/JAS.2021.1004207

[6] Statista. (2020). Internet user penetration in Nigeria from 2015 to 2025. Retrieved 19 September, 2020, from https://www.statista.com/statistics/484918/internet-user-reach-nigeria/#:~:text=This%20statistic%20provides%20information%20on,to%2065.3%20percent%20in%202025.

[7] https://www.vanguardngr.com/2020/11/why-nigeria-must-take-internet-development-seriously/

[8] https://www.ncc.gov.ng/documents/880-nigerian-national-broadband-plan-2020-2025/file

[9] https://guardian.ng/technology/leapfrogging-internet-infrastructure-deficit-for-nigerias-socio-economic-development/

[10] https://www.ncc.gov.ng/documents/976-challenges-of-technology-penetration-in-an-infrastructure-deficit-economy-nigeria-perspective/file pages 3-8

[11] https://nairametrics.com/2019/09/24/nigeria-needs-100-billion-annually-to-fix-infrastructural-deficit-finance-minister/

[12] Zheng, W., Zheng, Z., Chen, X., Dai, K., Li, P., & Chen, R. (2019). NutBaaS: A Blockchain-as-a-Service Platform. *IEEE Access*, 7, 134422-134433, 2019. doi: 10.1109/ACCESS.2019.2941905

[13] Satoshi, N., (2018). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved 2 June, 2018, from https://bitcoin.org/bitcoin.pdf.

[14] Jang, H., Lee, J. (2018). An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information. IEEE Access, 6, 5427-5437. doi:10.1109/ACCESS.2017.2779181

[15] Suzuki, S. & Murai, J. (2017). Blockchain as an Audit-able Communication Channel. *2017 IEEE 41st Annual Computer Software and Applications Conference*, 516-522.

[16] Catalini, C. & Gans, J. (2016). Some Simple Economics of the Blockchain. MIT Sloan School of Management, 44(12), 5191-16.

[17] Deloitte.com. (2017). Deloitte.com. Retrieved 20 November, 2017, from https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Technology/IE_C_BlockchainandCyberPOV_0417.pdf

[18] Dannen, C. (2017). Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners. Brooklyn, New York, USA: Apress

[19] Ansif, A., & Mohsin, R., (2016). Electronic voting with biometric verification offline and hybrid evms solution. *The Sixth International Conference on Innovative Computing Technology(INTECH 2016*), 332-337.

[20] Cortier, V., Constantin, C.D., Francois, D., Benedikt, S., Pierre,Y.S., & Bogdan, W.(2017). Machine-Checked Proofs of Privacy for Electronic Voting Protocols. *2017 IEEE Symposium on Security and Privacy*, 993-1008.

[21] https://www.diva-portal.org/smash/get/diva2:1111497/FULLTEXT01.pdf

[22] https://cse.sc.edu/~mgv/csce190f18/diPierro_mcs2017050092.pdf

[23] Sato, M., Matsuo, S. & Di matteo, T. (2017). Long-term public blockchain: Resilience against Compromise of Underlying Cryptography. *IEEE* ,1-8.

[24] Wei, C.C.J & Wen, C.C. 2018. Blockchain-Based Electronic Voting Protocol. International Journal On Informatics Visualisation. 2(4), pp. 336-341.

[25] Sankar, L. .S, Sindhu , M & Sethumadhavan, M. (2017). Survey of Consensus Protocols on Blockchain Applications. *2017 International Conference on Advanced Computing and Communication Systems(ICACCS -2017*), 1-5

[26] Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *2017 IEEE 6th International Congress on Big Data*, 557-564

[27] Ocáriz Borde, H. S. (2022). An Overview of Trees in Blockchain Technology: Merkle Trees and Merkle Patricia Tries. Researchgate.

[28] Apeh, J. A., Ayo, C. K., & Adebiyi, A. (2021). A latency-improved blockchain implementation model for nation-wide electronic voting system. Journal of Theoretical and Applied Information Technology, 99(22).