Application of Blockchain Technology for Electronic Voting

MIDN 1/C Chase Lee
Cyber Sciences Department
United States Naval Academy
Annapolis, United States of America
m223804@usna.edu

MIDN 1/C Jack Murray
Cyber Sciences Department
United States Naval Academy
Annapolis, United States of America
m224740@usna.edu

MIDN 1/C Luke Harkins
Cyber Sciences Department
United States Naval Academy
Annapolis, United States of America
m222736@usna.edu

***Abstract*—Our applied research delves into the use of the Ethereum blockchain technology and explores its application for electronic voting. The research question that we intend to address is, by researching the techniques of, and building a model for, a blockchain network for online voting, can we show the viability of blockchain technology, both in theory and security, for use in future elections? Our goal is to develop the model and evaluate whether it has the potential to be trusted by a respective constituency from both a social and technical perspective.**

I. Background

Abigail Johnson, the Chief Executive Officer of Fidelity Investments, has said: "Blockchain technology isn't just a more efficient way to settle securities. It will fundamentally change market structures, and maybe even the architecture of the Internet itself [1]." Although blockchain technology has recently been aggrandized, it certainly has implications that will greatly benefit and revolutionize current systems.

There are various definitions of blockchain. However, for clarity the following definition has been provided: "a blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain [2]." The cryptocurrency that sparked public interest in blockchain technology was Bitcoin. Bitcoin was created in 2009 after the economic crisis in 2008. However, what many people do not realize is that blockchain technology was first conceived in 1991 by Stuart Haber and W. Scott Stornetta, who sought to build a technology that could prevent document timestamps from being modified. Blockchain achieved its current state of infamy due to Satoshi Nakamoto, Bitcoin's pseudonymous creator, and the Bitcoin protocol. Nakamoto referred to the digital currency as "a new electronic cash system that's fully peer-to-peer, with no trusted third party [3]."

After Bitcoin was unveiled, other developers realized that Bitcoin only used the blockchain to record a digital transaction through a ledger. However, the blockchain can "be used to immutably record any number of data points [3]." The question that a non-technical person is likely to ask is, how can there be a guarantee that the ledger never changes? Each node, or for the purposes of applying blockchain technology to electronic voting, a person, owns a copy of the ledger. When a transaction is made, it notifies every node and updates their respective ledger. In other words, the technology implements a gossip protocol where each node sends out data to other nodes. Therefore, there is not a central place where the transaction exists because every node has access to that exchange.

Another important characteristic of this technology, especially in the context of cryptocurrencies, is that it is fungible. Fungibility describes an asset that can be traded for another asset of the same type of class. Crypto's "fungibility makes it a trusted means of conducting transactions on the blockchain [4]" because the value of a cryptocurrency will always equal in value the same aforementioned cryptocurrency, which allows for trading and exchanging for another.

Blockchain technology is designed in such a way that blocks are appended only, meaning that one cannot change a block, but only add new blocks to the blockchain. Every block contains a hash of the block that precedes it, which carries forward, like a linked list. The properties of hash functions make it computationally prohibitive to object fake history since every block verifies its precedent.

An example of a blockchain implementation is Ethereum. Ethereum is an open-source blockchain that follows the decentralized model and supports smart contract functionality. Smart contracts provide a service that will automatically execute once specific preconditions are met. Smart contract functionality is the use of programs that are stored on the blockchain. These programs do not require any intermediary's involvement and thus do not lead to any time loss that is created when relying on the fallibility of humans. A decentralized application, otherwise known as a dApp, utilizes a smart contract and frontend user interface, to provide a service.

II. Literature review

A. Properties of Blockchain Technology

As previously mentioned, blockchain technology has specific properties that make it better in comparison to the current methodologies which are used to carry out elections. One of these properties is the structure. Unlike a typical database that breaks down the data into tables, the blockchain will structure the data into blocks that are tied together. One of our desired properties in the context of election systems is that our system ensures the existence of an irreversible timeline of data. This is provided by the decentralized nature of blockchain protocols [3].

Furthermore, Fig. 1 below shows the important characteristics of Distributed Ledger Technology (DLT), which encompasses blockchain technology. These include programmable, secure, anonymous, unanimous, distributed, immutable, and time-stamped properties.

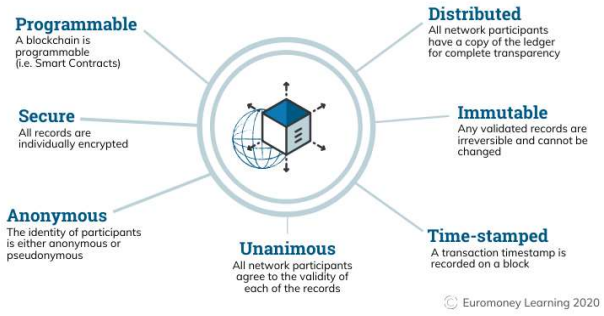

Fig 1. Properties of Distributed Ledger Technology (DLT) [5]

B. Decentralization

In the context of blockchain, decentralization is the key aspect to ensuring security of the electoral process. Blockchain technology is widely distributed and is not concentrated in one singular location. Therefore, each of the nodes hold its blockchain in different locations and operate under various organizations.

Each node, which is a computer on the network, stores the blockchain and its full record, also known as the digital ledger. By employing decentralization, if one node has an error, it could check the records of other nodes to correct itself. This prevents that node from changing anything on the digital ledger that it holds because the other nodes will offer a valid record.

It should be noted that any change that occurs to the system must have a majority of the network to agree on said changes. Whether those changes are nefarious, or for the benefit of removing false information, these changes would have to occur in the best interest of the majority.

C. Public, Consortium, Federated, and Private Blockchain

Public blockchain does not have a central authority, which governs the state of the blockchain. As the name implies, the public blockchain has no restrictions on participants and thus data on the blockchain can be accessed by anyone. However, a requirement for a public blockchain, like Bitcoin and Ethereum, to function is a digital currency as an incentive. Another consideration is the fact that public blockchains require high electricity consumption because it "results from the frequently used 'proof of work' algorithm for consensus building. Put simply, participants of a blockchain buy themselves lottery tickets by means of computing power. The owner of the winning ticket may next make a proposal for new data in the blockchain. If the proposal is accepted, the owner of the winning ticket will receive a reward paid in crypto currencies [6]."

Consortium blockchain, otherwise known as federated blockchain, only accepts certain participants. Each participant has equal power. "Since only pre-selected participants are allowed to participate, no direct financial incentives are necessary. Participants will be verified in advance and excluded in case of malicious behavior by the other participants [6]." Federated blockchains offer specific advantages over public blockchains. These advantages include speed, throughput, privacy, and energy efficiency.

Finally, the private blockchain has a node/person that controls the blockchain. For the most part, they are used to safeguard an organization's sensitive data because without proper authorization, they cannot use the network.

D. Ethereum

In 1996, Nick Szabo, a world-renowned computer scientist who began discussing smart contract technology published an article on smart contracts [7]. In layman's terms, smart contracts are similar to conditional transactions such that it upholds the IF X, THEN Y condition. From this article, the Ethereum team developed a platform to build smart contracts in a decentralized manner.

Many people are familiar with Bitcoin because it sends monetary value between people. However, Ethereum differentiates itself by sending information between programs. Ethereum would allow developers to write applications and programs to execute in a decentralized environment.

It is important to note the distinction between ether and Ethereum. Ethereum is a platform and ether is the main currency that is used on the platform. Ether is necessary to operate the decentralized, distributed platform. "Ether is the incentive ensuring that developers write quality applications...and that the network remains healthy (people are compensated for their contributed resources) [8]."

E. Proof-of-Work

In order for blockchains to function successfully and securely, the Proof-of-Work algorithm, also known as consensus algorithms, limit new transactions on blockchain for only legitimate users. In the case of Proof-of-Work, nodes compete against each other to solve the computations, which is a process known as mining because a significant amount of resources are needed to solve the problem. Once a node

completes the computation before others, it can add a block to the network and broadcast the new block to the other nodes. Once the other nodes have verified the legitimacy of the new block, it will add itself to the blockchain network. As a result, the node is given a specified amount of cryptocurrency in exchange for its services [9].

F. Applying Blockchain to Elections

Unlike physical ballots, it is computationally infeasible to shred blocks of the blockchain. As discussed previously, since one block contains its preceding block's hash, it will not register if its respective block does not contain the correct hash of the previous block. Similarly, if a person or organization wanted to alter the results, it is computationally infeasible because new blocks are stored linearly and chronologically. Editing a single block would require editing every single block after it.

One methodology to approach blockchain enabled electronic-voting (BEV) is issuing users "wallets," which provide the authentication that gives each person a coin, or in our application, a ballot. The public digital ledger, in other words the blockchain, approves and adds the verified vote, given that the user has been authenticated. More sophisticated authentication schemes, like multi-factor authentication can be implemented.

Using blockchain for voting is not a new idea. It has been used by a South Korean province and an Estonian technology company that have already begun testing this process. It has shown to prevent any form of tampering, provides greater transparency, less ambiguity, increased vote tallying speed and cost effectiveness. A potential drawback is that the database of verified credentials and validation procedures is significantly larger when constructing an election on a national scale.

G. Online Voting Application Using the Ethereum Blockchain

We have generically discussed the use of blockchain technology for electronic voting. However, let us evaluate the potential benefits of using the Ethereum protocol. In Ethereum, each account can be identified by a 20 byte address. Each account contains a nonce, which is a counter to keep track of transactions, "ether" balance, "contract code," and account "storage." Therefore, the central authority, which in the case of national elections would be the federal government, would create a unique identifier for each verified constituent. The user's vote would be encoded in an Ethereum transaction. That vote is then encrypted by the user's private key and verified by their respective public key, which facilitates their vote being added to the digital ledger or blockchain. The transaction containing the user's vote is then digitally signed using the ECDSA algorithm and the user's private key. The digital ledger uses a concoction of public key and transaction methods to verify what has occurred in terms of transactions [10].

In regards to the application on a national level, the government cannot solely control the framework, otherwise it will not provide the transparency that was intended. The databases used need to be available to keep the government, or the central authority, from dictating and tampering the blockchain.

H. FollowMyVote.com

Over the past decade, around 2005 to 2007, some countries have explored electronic voting implementation, e.g. Estonia and the Netherlands. Therefore, organizations have begun development of decentralized applications like "Follow My Vote [which] has created end-to-end verifiable online voting software that is open-source [11]." Follow My Vote uses a blockchain-based voting system. The voter installs the voting booth application on their device. The voter submits their identity information to verify that they are approved to vote. Once verified by the appropriate identity authority, the voter can request a ballot, which is issued by the Registrar. The voter submits their respective vote to the blockchain ballot box. To obtain proof of ballot, the voter has a receipt of their participation. Follow My Vote even allows the voters to change their voting prior to the close on Election Day. The voters are able to audit each ballot in the ballot box while maintaining anonymity of those who voted.

When the polls close on Election Day, the most current votes submitted by each voter would be considered the official votes. Voters can follow their vote into the ballot box to ensure that their vote was cast as intended and counted as cast. If they choose to do so, each voter would also be allowed to audit each ballot in the ballot box to confirm the vote totals being reported by our blockchain voting system are accurate, without revealing the identity of each voter. "At Follow My Vote, [they] want every voter to have faith in the democratic process, trust in their government, and feel like their voice matters [11]."

I. Paillier Cryptosystem

In 1999, Paillier proposed a cryptosystem that utilized a homomorphic encryption scheme. Homomorphic encryption allows the developer to perform operations on the encrypted data. For instance, numbers n_1 and n_2 are encrypted using the public key encryption scheme. The ciphertexts that are produced include $c_1 = E_{public}(n_1)$ and $c_2 = E_{public}(n_2)$. Homomorphic encryption is unique such that computations can occur directly to encrypted numbers without needing access to the private key.

Homomorphic properties enable the addition of responses in the encryption scheme. This means that any individual can perform the addition function with c_1 and c_2 after decrypting the sum of the numbers:

$$D_{private}(addition_{public}(E_{public}(n_1), E_{public}(n_2))) = n_1 + n_2 \text{ [12]}$$

This algebraic property allows the developer "to compute with encrypted values without knowing the content of ciphertexts. They are useful when the anonymity of users is required. For example, in an election with multiple candidates, a single voter's response will encrypt the entirety of the

candidate tallies for that ballot rather than just the candidate who received the vote. Furthermore, since this entire tally can be used on the whole election system, it allows for the tally to be computed without decrypting one vote at a time and therefore guaranteeing the privacy of users [13].”

J. *Web3.py*

In order to interact with Ethereum, Web3 Python provides a useful library which can be used to send/receive. This is commonly used as a tool for decentralized applications. However, for the purposes of this research, it will be used to send and receive transactions. Web3.py is derived from Web3.js, which is a JavaScript API [14].

III. Proposed Methodology

A. *Dependencies*

Go Ethereum:

Go Ethereum is an original implementation of the Ethereum protocol. Go Ethereum is written in Go, fully open source and licensed under the GNU LGPL v3 [14]. Using a C compiler and a Golang implementation, it builds geth.

B. *The Decentralized Application Implementation*

Go-Lang:

The consensus protocol is the key component of any blockchain network because all the nodes must reach a necessary agreement about the state of the distributed ledger. By modifying the consensus algorithm in consensus.go, it makes the difficulty algorithm easier for a new block to be created at time given the parent block’s time and difficulty.

The mining difficulty was set to the lowest possible level because it allows a large amount of faux currency to be mined for transactions between accounts.

```
// CalcDifficulty is the difficulty adjustment algorithm. It returns
// the difficulty that a new block should have when created at time
// given the parent block's time and difficulty.
func CalcDifficulty(config *params.ChainConfig, time uint64, parent *types.Head
er) *big.Int {
        /*next := new(big.Int).Add(parent.Number, big1)
        switch {
        case config.IsArrowGlacier(next):
                return calcDifficultyEip4345(time, parent)
        case config.IsLondon(next):
                return calcDifficultyEip3554(time, parent)
        case config.IsMuirGlacier(next):
                return calcDifficultyEip2384(time, parent)
        case config.IsConstantinople(next):
                return calcDifficultyConstantinople(time, parent)
        case config.IsByzantium(next):
                return calcDifficultyByzantium(time, parent)
        case config.IsHomestead(next):
                return calcDifficultyHomestead(time, parent)
        default:
                return calcDifficultyFrontier(time, parent)
        }*/
        return big.NewInt(1)
}
```

Fig 2. Update the Mining Difficulty of Consensus.go

Genesis JSON Block:

The genesis block is the start of the blockchain, and this JSON file will define settings for the blockchain network. For instance, it specifies the difficulty of mining blocks as well as the parent hash.

```
navy@navy-VirtualBox:~/gethDataDir$ cat genesis.json
{
        "config": {
                "chainId": 7986,
                "homesteadBlock": 0,
                "eip150Block": 0,
                "eip155Block": 0,
                "eip158Block": 0
        },
        "alloc": {},
        "coinbase"      : "0x0000000000000000000000000000000000000000",
        "difficulty"    : "0x5",
        "extraData"     : "",
        "gasLimit"      : "0x2fefd8",
        "nonce"         : "0x0000000000000042",
        "mixhash"       : "0x0000000000000000000000000000000000000000000000000000000000000000",
        "parentHash"    : "0x0000000000000000000000000000000000000000000000000000000000000000",
        "timestamp"     : "0x00"
}
```

Fig 3. Creation of the Genesis JSON Block

After running “geth –datadir ~/gethDataDir init genesis.json,” the chain with the genesis json block is initialized.

Open IPC Endpoint:

After running “geth –datadir ~/gethDataDir –networkid 7986,” the host is connected to the Ethereum private blockchain, with network ID 7986, which is also noted in the genesis block.

```
Terminal                Jan 19  16:04
navy@navy-VirtualBox: ~/gethDataDir
navy@navy-...   navy@navy-...   navy@navy-...   navy@navy-...
tried=112 static=0
INFO [01-19|16:02:27.606] Looking for peers                        peercount=0
tried=84  static=0
ERROR[01-19|16:02:34.965] Snapshot extension registration failed   peer=7f17c61
9 err="peer connected on snap without compatible eth support"
INFO [01-19|16:02:37.628] Looking for peers                        peercount=0
tried=102 static=0
INFO [01-19|16:02:47.636] Looking for peers                        peercount=0
tried=68  static=0
INFO [01-19|16:02:59.652] Looking for peers                        peercount=0
tried=80  static=0
INFO [01-19|16:03:09.762] Looking for peers                        peercount=0
tried=127 static=0
INFO [01-19|16:03:19.982] Looking for peers                        peercount=0
tried=90  static=0
INFO [01-19|16:03:30.003] Looking for peers                        peercount=0
tried=136 static=0
INFO [01-19|16:03:40.011] Looking for peers                        peercount=0
tried=78  static=0
ERROR[01-19|16:03:41.024] Snapshot extension registration failed   peer=f3aa604
6 err="peer connected on snap without compatible eth support"
INFO [01-19|16:03:50.142] Looking for peers                        peercount=0
tried=86  static=0
INFO [01-19|16:04:00.155] Looking for peers                        peercount=1
tried=132 static=0
```

Ubuntu Software

Fig 4. Opening IPC Endpoint and Connecting to Ethereum Network

Attach IPC Endpoint:

After running “geth attach ipc:gethDataDir/geth.ipc,” this allows the user to input JavaScript commands in the console. This allows the administrator to create an account for each verified user on the network. This also allows the administrator to create the coinbase account. Additionally, the

JavaScript console can be used to view the results of the election.

Fig 5. Attaching the IPC Endpoint Terminal

Setting Up the Mining Account:

After attaching to the network, the administrator would run "web3.eth.coinbase" in the JavaScript console. The account that is printed to the console is the address account in which mining will be rewarded and deposited.

Fig 6. Creating the Coinbase Account

Setting Up the User for Voting:

After attaching to the network, a new account is created for the user. This is achieved using the command "web3.personal.newAccount('test123')." Note that 'test123' is the password for this account. Although not addressed in this paper, the password would be RSA encrypted in order to protect the confidentiality of the user.

Fig 7. Creating a User Account to Vote and View Results

HTTP RPC Listener:

After running the "admin.startRPC('127.0.0.1',8545, '*', 'web3,net,eth,personal,debug,miner')," the method creates an HTTP based JSON RPC API webserver to handle client requests.

Python Ballot File:

The first step is to unlock the user's account as well as the coinbase account.

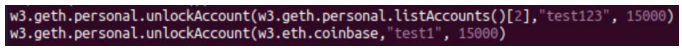

Fig 8. Unlocking the User's Account and the Coinbase Account

The second step is to use the Python 3 library for Partially Homomorphic Encryption using the Paillier crypto system. The public and private keys are generated as a keypair to protect the confidentiality of the voters.

IV. Implementation

A. Assumption for Testing Implementation

The election script was designed for testing purposes. Therefore, this user has the ability to send multiple transactions, which is the equivalent of voting for multiple individuals. This account could be used as an administrator account, but in actuality, the account would be restricted to a singular ballot vote per account.

B. Voting Functionality

In order to send the transaction, the local private and public keys are generated for the user. The public key is then used to encrypt the tally of votes for the specific candidate. The candidate's name is separated from the Paillier encryption tally of votes with the use of a dash. This message is then converted into UTF-8 and sent from the user account to the coinbase account. The input field within the transaction contains this encrypted data. In order to send the transaction, a certain amount of time is allocated towards mining on the network.

Additionally, the user is able to add a new candidate or choose from those that are already added to the ballot. The figure below shows the vote of Candidate 1 and returns the transaction.

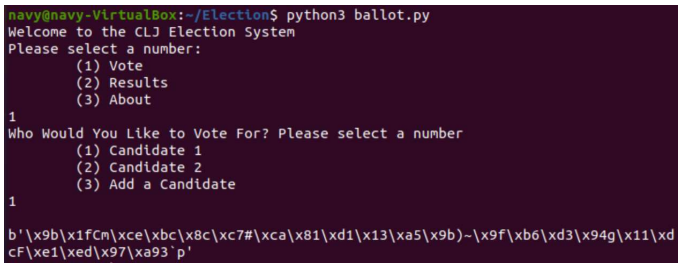

Fig 9. Voting for an Already Specified Candidate

The figure below shows the new candidate who the user would like to vote for.

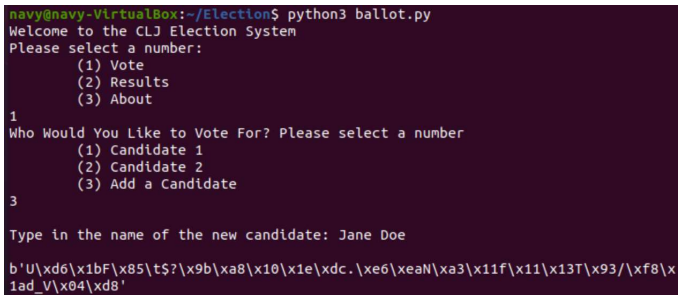

Fig 10. Voting for a New Candidate

For the purposes of testing functionality, the figure belows demonstrates the result of the election after the user would have voted for Candidate 1 and Jane Doe.

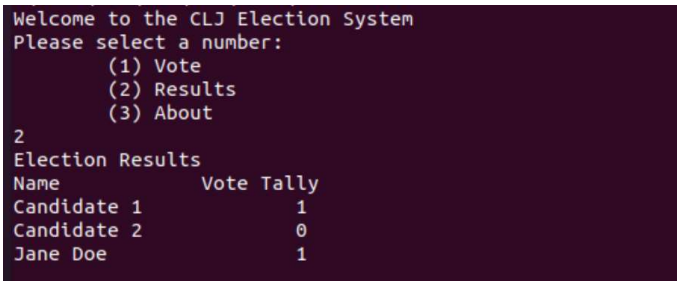

Fig 11. Results of a Test Election for Candidate 1 and Jane Doe

C. *Result Viewing Functionality*

In order to view the results, the latest block is obtained from the network. Since the mining difficulty is set to the lowest extent, some blocks will not contain any transactions. Therefore, after iterating through the recent transactions, each block will have their respective transactions checked. The input field in a non-empty transaction will contain the candidate's name and the encrypted Paillier value of their tallied votes.

As shown in the previous figure, the results are then collected in a dictionary after being decrypted using the Paillier private key.

V. Analysis

A. *Technical Assumptions for the Implementation of the Election System*

There were several assumptions made because our research only sought to build an election system Proof of Concept. In regard to the authentication and authorization of valid users, this aspect was already assumed. The final product used hard-coded accounts that were authorized by the network. In the event that a new user needed to be added to the system, it would simply require more steps to build the backend support to add a new account with the desired password.

Similarly, another assumption was the security of the password tied to each account. In this case, the password was not protected with any form of encryption. However, it is assumed that additional encryption schemes would be employed to protect the confidentiality of the user's account. This consideration was outside the scope of the research. Nevertheless, it would be simple to implement with an encrypted password on the Ethereum network after encrypting on the client-side. Further research would intend to find the best methodology to protect the private and public keys. The public key would likely be written to the network and the private key would be generated to protect the confidentiality of the user's password and thus their vote. Keeping this in mind, a secure password would only serve to prevent malicious users from casting a vote. It would not allow the malicious user to read the valid user's vote since the valid user's ballot is automatically encrypted using the Paillier cryptosystem.

In addition, an arbitrary supply of ether is used and mined every time that the user desires to write, in other words cast their ballot, to the network. The private test network built in this case made it easy in terms of difficulty to mine. As explained previously in the Genesis JSON Block analysis, this allowed for the user to cast their vote in a timely manner.

B. *Societal Considerations for Development of Blockchain Technology for Electoral Process*

While we have shown the numerous facets of a blockchain voting system, the social impact and acceptance of e-voting is just as important when discussing the overall success of online voting. For example, in a 2005 article on teledemocracy, researchers found that the most important feature of an e-voting system was the inclusion of a public list of procedural security measures [15]. In order to build trust in e-voting systems, users required a non-technical understanding of the built in security standards in an e-voting. This "blind faith" is similar to the trust passengers place in the standards that control aircraft operations despite the fact they don't understand the science or the technical procedures themselves.

In addition, issues on ballot secrecy and verifiability are other major concerns when considering social impact/acceptance of e-voting. For example, a 2013 study found that voters who submitted ballots via e-voting were less confident than their paper ballot counterparts that their votes had been counted. This creates a threat to the social acceptance of large-scale e-voting unless a verification process can be introduced where users can be ensured that their vote has been counted correctly [16].

With that being said, the confidentiality of ballots on e-voting systems has been found to be most detrimental to the acceptance of e-voting. In the same 2013 study on the impact of e-voting, it was found that there was a seven percent difference in the trust voters had that paper ballots would be more confidential than an electronic ballot [16]. As a result, if e-voting is to gain acceptance, there will need to be publicized security procedures as well as increased trust in the confidentiality and verifiability of e-voting.

Acknowledgment

The authors gratefully acknowledge the contributions of T. Mayberry and LT V. Kanth, USN, to the research and its findings.

References

[1] P. Rizzo. (2017, May). “Fidelity CEO Talks 'Love' For Bitcoin, Why Blockchain Will 'Change' Markets.” *CoinDesk* [Online]. Available: https://www.coindesk.com/markets/2017/05/23/fidelity-ceo-talks-love-for-bitcoin-why-blockchain-will-change-markets/

[2] EuroMoney Learning. *What is Blockchain?* Available: https://www.euromoney.com/learning/blockchain-explained/what-is-blockchain

[3] Investopedia. *Blockchain Explained.* Available: https://www.investopedia.com/terms/b/blockchain.asp

[4] R. Conti and J. Schmidt. (2021, May). “What You Need To Know About Non-Fungible Tokens (NFTs).” *Forbes* [Online]. Available: https://www.forbes.com/advisor/investing/nft-non-fungible-token/

[5] Phemex. *What is Distributed Ledger Technology.* Available: https://phemex.com/academy/what-is-distributed-ledger-technology-dlt

[6] Origin Stamp. *Public vs. Consortium vs. Federated vs. Private Blockchain*. Available: https://originstamp.com/blog/public-consortium-private-blockchain/

[7] M. Gord. (2016, April). “Smart Contracts Described by Nick Szabo 20 Years Ago Now Becoming Reality.” *Bitcoin Magazine* [Online]. Available: https://bitcoinmagazine.com/technical/smart-contracts-described-by-nick-szabo-years-ago-now-becoming-reality-1461693751

[8] C. Burniske and J. Tatar, *Cryptoassets: The Innovative Investor’s Guide to Bitcoin and Beyond*. McGraw Hill Education, 2018.

[9] L. Conway. “Proof-of-Work vs. Proof-of-Stake: Which is Better?” *Blockworks* [Online]. Available: https://blockworks.co/proof-of-work-vs-proof-of-stake-whats-the-difference/.

[10] V. Buterin. *Ethereum Whitepaper*. Available: https://ethereum.org/en/whitepaper/.

[11] Follow My Vote. *Blockchain Voting: The End to End Process.* Available: https://followmyvote.com/blockchain-voting-the-end-to-end-process/

[12] OpenMined. (2020, July). *What is the Paillier Cryptosystem?* Available: https://blog.openmined.org/the-paillier-cryptosystem/

[13] O. Baudron, P.-A. Fouque, D. Pointcheval, G. Poupard, and J. Stern (2001, August). “Practical Multi-Candidate Election System.” *Proceedings of the 20th ACM Symposium on Principles of Distributed Computing*. Available: https://people.csail.mit.edu/rivest/voting/papers/BaudronFouquePointchevalPoupardStern-PracticalMultiCandidateElectionSystem.pdf

[14] Web3 Python. *Introduction Documentation*. Available: https://web3py.readthedocs.io/en/stable/

[15] Go Ethereum. *Official Go Implementation.* Available: https://geth.ethereum.org/

[16] Xenakis, Alexandros and Ann Macintosh. “Procedural Security and Social Acceptance in E-Voting.” *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (2005): 118a-118a.

[17] R. Michael Alvarez, Ines Levin, Julia Pomares and Marcelo Leiras (2013). “Voting Made Safe andEasy: The Impact of e-voting on Citizen Perceptions.” *Political Science Research and Methods*, 1,pp 117-137 doi:10.1017/psrm.2013.2