www.ijarp.com ISSN 2456-9992 Page: 01-14

International Journal Advanced Research Publications

Research Article

Volume: 02
Issue: 08

BLOCKCHAIN-BASED ELECTRONIC VOTING SYSTEM

***Thin Zar Ko Ko**

Faculty of Computer Science, University of Computer Studies, Hinthada.

Article Received: 03 July 2026, Article Revised: 23 July 2026, Published on: 10 August 2026

***Corresponding Author: Thin Zar Ko Ko**

Faculty of Computer Science, University of Computer Studies, Hinthada.

Doi: https://doi-doi.org/101555/ijarp.2725

ABSTRACT

It has been a problem for a long time to create electronic voting (e-voting) system which meets the legal demands of politicians. Digital currency technologies have become an exciting development of technology in the field of information technology. Blockchain systems deliver an infinite variety of sharing economic applications. The purpose of this paper is to assess the use of blockchain as a tool for the implementation of decentralized e-voting system. This paper outlines criteria for creation e-voting system and the ethical and technical constraints that blockchain can be used as a tool for implementing those systems. The paper begins with an overview of several of the common blockchain system that blockchain provides as a software. I also suggest a new, blockchain-based e-voting method addressing all the constraints we have found. More broadly this paper assesses the ability of decentralized ledger technology by discussing a case study, i.e. the voting system and applying a blockchain-based framework that enhances safety which reduces costs of organizing a national election.

KEYWORDS: Blockchain; E-voting; Digitalization; Election.

1. INTRODUCTION

Election security is a matter of national security in every democracy. In order to minimize the expense of conducting a national electoral process while meeting and growing security requirements of an election, the field of computer security has researched the possibilities of e-voting systems [12] for a decade. The voting system was built on pen and paper from the beginnings of the democratically elected candidates. To prevent abuse and to trace and verify the voting process, replacing the conventional paper and pen method with a modern election system is critical [15]. The security group, largely focused on physical safety issues, found

the e-voting machines to be faulty. Everyone with direct barriers to the system can decrypt the data and thus affect all votes cast on the computer referred to above. A blockchain is an immutable, distributed, public ledger and incontrovertible. Four main features of this new technology: (i) The ledger is accessible in many places: there is no single point where the distributed leader is managed. (ii) The authority of that can add latest ledger transactions are dispersed. (iii) The "new block" proposed in the ledger should refer to previous version of the ledger, create an ethernal chain from which the blockchain gets its name, and thus ensure that the credibility of previous ones is not damaged. (iv) Much of the nodes in the network must develop awareness until a new entry block is a permanent directory part. Such technology innovations work with advanced cryptographic, giving the same or/and larger level of security than any previously recognized database. Therefore, many including other perceive blockchain technology [17] as the perfect tool to build a new, modern and public vote. This article assesses use of such blockchain for e-voting system and includes the following basic developments: (i) study current blockchain systems designed to the development of an e-voting system based on blockchains, and (ii) suggested a blockchain e-voting system using "enabled blockchains" to allow for liquid democracy.

2. RELATED WORKS

I will review in this section numerous research articles and thesis discussing related subject areas of study, namely e-voting systems. The two-round public debate introduced an anonymous vote to introduce two-Round Anonymous Veto Protocol (called AV-net) to a self-computing system. The paper concentrated on the dining cryptograph network (DCnet) and vulnerabilities, thus proposing newly the A-V network to resolve this problem [6][5][18], compared to related techniques. Does not allow for reliable third party or private channel is provided by new structure, such as the AV-net. The protocol is carried out by the network by sending out public two-round messages, but the number of rounds, the cost of computation and the use of bandwidth are far more effective. The new protocol usually split e-voting into 2 classifications: 1) Decentralized elections where only voters are responsible for the process. 2) Centralized elections where the process is run by trusted administers. The proposed protocol tended to focus on first class, in which the primary objective, two problems, was strong voting confidentiality. The first problem was that a trustable third party does not exist. Many security problems can be solved easily by a trusted third party, but can lead to the "trustable" third party breaking the policies on security. That aim then was to eliminate it completely and using a trustworthy political party. Second problem had been that private

networks to ensure independence from conflicts would be no voter-to-voter, i.e. everyone must verify if all voters correctly followed the procedure. In AV-net, these problems have been met, but the proposed protocol is a different approach that addresses the negative effect of AV-net, which is a high calculation burden on any elector, which is linearly increased by the number of votes. The new protocol has proven to be as reliable as the AV-net, but can be considered a generalization of the AV-net protocol. The first round was to a report zero knowledge proof (ZKP) and a public key for their personal key for any participant in the two-round protocol. The ZKPs shall be verified by each participant and the round shall be calculated after completion. In the second round, each participant would show, without disclosing which one, that encrypted voting was one of the most valid votes. The self-decoding mechanism then functions in such a way that the public keys of all members are combined that the random elements of the private key disappear instantly, exposing the decrypt. There are limits on national elections under this protocol. The system needs all the voters to work together, otherwise the protocol would not work. Each individual, for example, will know who voters are and may eject them if any voter refuses to send data at the end of round 1 and restart the procedure. The tallying process will fail. This leads to a delay of the voting process, which is extremely expensive for a major vote. The lack of resistance to intimidation due to the remote voting, which can compel an elector to disclose their hidden value and thus expose his voting methods, is another weakness. The first project to introduce the decentralized and autonomous internet voting protocol with the full voting privacy, called The Open Vote Network (OVN) for boardroom voting, was the Smart Contract for Boardroom Voting on Maximum Voter Privacy [13]. OVN is written to the Ethereum blockchain as a smart contract. In its main concept, the OVN carries out the anonymous vote through a two-round public discussion that discussed previously. Developers of OVN estimated after the system had been introduced that it would cost $0.73 per user to run the system on the Ethereum blockchain. The maximum safe limit of 50 electors was considered reasonable, as the maximum privacy of voters was provided and it can be publicly verified. Due to the gas cap on the public Ethereum blockchain the restriction of the number of voters was recommended. The weaknesses of the previous protocol remain unchanged when reviewing their study paper. In the way that the vote is carried out in an unauthorized environment, the OVN doesn't provide coercion with public verification, i.e. the coercer stands above the voter's shoulder. The OVN itself is weak to denial-of-service attacks since the Ethereum blockchain, which performed several DOS attacks during its existence, is introduced. In the case of an election, a large number of transactions could also be made in

the blockchain of Ethereum, which could greatly postpone voting. Therefore, the implementation could be ideal small boardroom voting with every voter downloading the entire blockchain of Ethereum in order to validate the voting protocol has been correctly implemented.

3. APPLICATION OF BLOCKCHAIN

Blockchain technology was implemented when the first cryptocurrency called Bitcoin was developed by Satoshi Nakamoto in 2008. In conjunction with the PoW (Proof-of-Work) stochastic consensus protocol, Bitcoin technology uses a decentralized public ledger, offering economic benefits to document the blockchain series in full order. At any transaction, the chain is repeated, encrypted, and publicly verifiable so that the data which is written on the blockchain are not interfered with by anybody. The blockchain structure is a data structure which can be appended only to it, so that new data blocks can be added but not changed or removed. The blocks are chains such that each block is hashed to ensure immutabilities, which is a function of the previous block. Generally, the Bitcoin blockchain publishes all elements of the entire chain, but other blockchains can be private, consortium-based or public. Public blockchains give read access to any user of this network and the ability to create a transaction. It is primarily used in cryptocurrencies (for example, Auroracoin, Dogecoin, Ethereum, and Bitcoin). Blockchain Consortium is a blockchain [4] "partially decentralized," where a pre-selected group of nodes manages the consensus method. Assume a consortium of 15 financial firms operating each with a node, 10 of which have to sign each block to ensure the block is valid. The participants may have the right to read the blockchain or to be limited. Not only does private blockchain restrict write access, but also read access to individual users who are able to validate their operation privately. This makes the operation on a local network less expensive since only few trusted and highly processed nodes have to be verified. Nodes can be connected exceptionally well and defects can be easily corrected by manually action, so that consensus algorithms can be used, which are final after much shorter block times [4]. We will use in our paper an approved blockchain, a consortium-based chains variation that uses a consensus algorithm for proof-of-authority (POA). Blocks and transactions are authenticated by validators, known as authorized accounts, in proof-of-authority-based networks. This is an automatic procedure which needs no regular monitoring of the machines by the validators. An accepted POA cryptographic algorithm blockchain helps one to randomly check and authorize the blockchain and censor transactions, and their credibility and identification, restrictions on a collection of selected known entities. Miners

using the proof-of-work consensus algorithm on a decentralized blockchain should otherwise do this. Anyone validating the system is charged by acting as validators in the system for the service they provide, rather than by paying mining fees, such as the public blockchains which are in use. Furthermore, an unauthorized person may track traffic or read input data through a private network. This is important in order for voters to cast ballots without revealing their identification or voting data.

- ***Smart Contracts*** - Smart contract applications running in a decentralized (e.g., blockchain) system are trackable and immutable. Nobody may modify the code or alter its execution actions once the intelligent contract has been deployed. Smart contract compliance assures binding parties to a written document. This generates a different kind of strong faith that is not focused with one group. Smart contracts allow for better control of digital contracts because they self-executing and self-verifying [16]. Nick Szabo, who graduates in law and computer science, earns the term and definition of "smart contracts". With smart contracts, his aim is expressed in the design of electronic commerce protocols between foreign persons on the internet through highly developed practices of law. Ethereum offers a blockchain open-source framework to build smart contracts. To write such contracts, Ethereum has introduced a new programming language named Solidity (similar to JavaScript). As a smart contract-based system in a blockchain enabled, I deploy my e-voting system.
- ***Non-Interactive Zero-Knowledge proof*** - There is no verification of proof of another concept which is related specifically to blockchain and could be considered and important element in meeting some criteria for the construction of an e-voting system on a blockchain is zero-knowledge proof. A zero-knowledge proof is an encryption technique in which a participant, the prover, can proof that it knows X without knowing any information other than that which the verifier knows X. Konstantinos Chalkias and Mike Hearn showed it live for the first time [9]. The ZKP works as follows with the example of “Two balls and the color-blind friend”: Prover has two red and one green balls, which is the same otherwise. Color-blind is the checker (friend). You will submit your friend balls, hidden behind his back to show that they really are different in color. Your friend then determines whether or not to turn the balls, and then shows a ball. The prover declares if the balls were switched. The prover will show by continuing this approach can be correctly classified since the verifier ensures also that chances of success are regularly halved. A NIZKP for short or non-interactive zero-knowledge proof is a type of zero-knowledge proof in which the prover and verifier have no interaction. Blum, Feldman and Micali [10] demonstrated that a similar string shared in

between verifier and the prover is sufficient to obtain machine knowing without interaction. However, the heuristic Fiat-Shamir [1] showed that a random oracle can obtain NIZKP, a cryptographic hash function in practice [11] that can instead be used to prove its identity and validity without a common public key. It is also possible. This system is best designed for smart cards, computers and remote-control systems, such as micro-processor-based machines. Thus, Fiat-Shamir's heuristic is an easy, effective, and safe way to authenticate, verify and guarantee the privacy of voters for qualified individuals. In this paper, we analyze and assess the viability for the implementation of an existing e-voting scheme, blockchain based and non-blockchain. On this basis, we have developed an e-voting system based on blockchain which optimizes the needs and considerations identified. We will then analyze various blockchain structures that can be used to execute and enforce smart contracts in the following paragraph, describing the function and component of implementing an e-voting contract. In the last paragraph, the design and architecture of the proposed framework will be described.

3.1. Smart Contract Election

The concept of a smart contract involves specifying functions involved in the agreement and various elements and transactions in the process of the agreement. I begin by describing electoral functions and then the process of elections.

3.1.1. Election Roles

The elections in my paper allow individuals or organizations to participate in the following positions, as shown in Figure 1. Where many organizations and individuals may act in the same role.

1. ***Administrators of Elections*** - Manage the electoral lifecycle. This position is enrolled by many trusted institutions and firms. The administrators of election determine the form of election and establish the elections above, configure ballots, register voters, decide lifetime and allocate authorized nodes.
2. ***Voters*** - Voters will authenticate, load, cast and validate the voting after the elections are over, for the elections for which they are registered. Voters will be rewarded with tokens for voting when they vote during an election that could be built into a smart city project in the very near future.

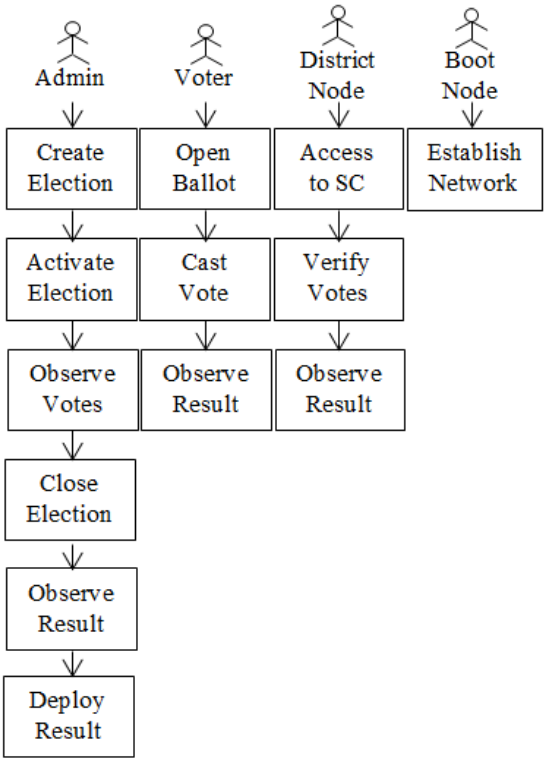

Figure 1. Roles and Procedures for Election.

1. ***Nodes of District*** - Could ballot smart contract representing could voting district will be put on a blockchain if election administrators establish an election. If smart voting contracts are established, each district node is allowed to communicate with its respective smart voting agreement. Where an elector casts its vote on its respective smart contract, all corresponding district nodes validate the voting data and when block time has been reached, each vote they agree on is added to the blockchain.
2. ***Bootnodes*** - Any institution hosts a bootnode with approved network access. A bootnode helps to discover and connect with the district nodes. Bootnodes do not carry the blockchain state and have a static IP, so that district nodes locate their peers faster [14].

3.1.2. Process of Election

In this paper, each electoral process consists of a set of smart contracts, which the electoral administrators instantiate on the blockchain. For each electoral district, a smart contract is specified such that several smart contracts are involved. The smart contract with the relevant voting area is requested to voter authenticates him after the user during the vote for every voter with appropriate polling discipline location specified in the registration stage. The main activities of the electoral process are as follows:

i. ***Creation of Elections*** - Election administrators build ballots using an app (dApp) which is decentralized. This decentralized app communicates with a smart contract election development agreement that establishes a list of candidates and polling districts by the administrator. This smart contract generates and deploys a set of smart voting contracts for each voting district, which includes a list of candidates, with a parameter in each smart ballot contract for every voting district. Each district node, when the election is held, is allowed to interact with the related vote smart contract (see Figure 2).

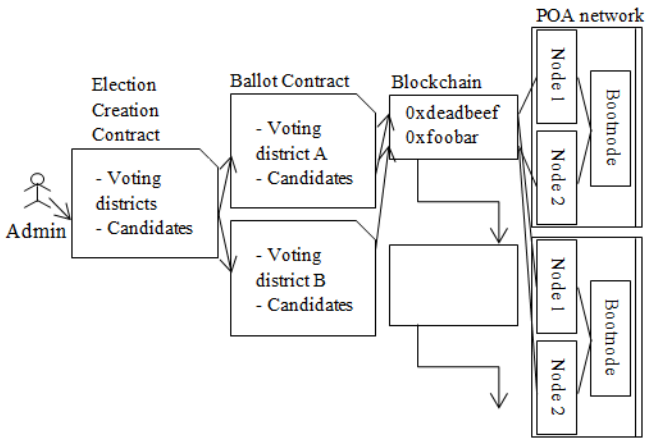

Figure 2. Election as a smart.

ii. Registration of Voters - Electoral administrators shall conduct the registration of voting process. When an election has been held a deterministic list of eligible voters must be specified by election administrators. It needs a portion to securely authenticate and approve qualified individuals for government identity verification service. The electronic ID, the PIN numbers, and details about the voting district the voter is located should be used by each eligible voter for such verifying services. A corresponding wallet will be created for each qualifying elector. For every election the voting party is eligible to vote in a single wallet can create an NIZKP. So that the process itself cannot consider which one vote wallet holds. It should generate such a wallet for every single elector.

iii. Transaction of Vote - When a vote is cast in a voting district, the voter communicates with a smart voting contract that is as specified for each voter in the voting district. This smart contract communicates with the blockchain by means of the appropriate district node, which applies the blockchain vote to it if there is a consensus between the majorities of the applicable district nodes. A vote is stored on the blockchain as a transaction, while for their verification purposes each individual voter receives a transaction ID. That blockchain transaction includes details about who voted for it and where the above vote was taken. Any

vote shall be appended to the blockchain by its respective voting smart contract only if the appropriate district nodes agree that the voting data are verified. The weight of a wallet decreases by 1 when an elector casts his vote, so that they are not allowed to vote by election more than once. As shown in Table 1, the transaction ID, the block placed in the process, the transaction age, the wallet sent and received by the public Ethereum blockchain, the total value sent and the transaction fee are all included in the transaction ID. All this information would not apply for a transaction in this paper. In this case N1SC means that the voting has been obtained from district N1. The only transaction information available to each transaction is the transaction ID, the block in which this transaction occurs. Ultimately the meaning of the transaction is the data that was chosen to cast, which reveals that the party D was the voting cast. Therefore, no information about individual voters who cast this specific vote is revealed in an operation in this paper (see Table 2). In order to protect voters against timing attacks, the age of a single transaction is excluded.

iv. Results of Evaluation - In smart contracts, the election are reported on the fly. The outcome of every smart contract vote is their own with its own storage for its corresponding location. The final result is released for each smart contract when the election is over.

v. Verified Voting - Every single voter receives a transaction ID of his or her vote, as stated earlier. After authentication through their electronic ID and their PIN, each elector will go and show his transaction ID to his official government. The official government uses blockchain node to find the transaction on the blockchain with the respective transaction ID. Blockchain explorer uses the blockchain official. The elector will thus see his blockchain vote, verify it has been correctly tallied and tallied.

Table 1. Example of a public transaction. (Ethereum)

TxHash	Block	Age	From	To	Value	[TxFee]
0xdead...	1337	33 sec ago	0xbeef...	Token	10 Ether	0.087
0xface...	1337	33 sec ago	0x4242...	0x1234...	1 Ether	0.056

Table 2. Example of a transaction in this paper.

TxHash	Block	To	Value
0xdeadbeef...	1337	N1SC	D
0xG1345edf...	1330	N2SC	P

Table 3. Framwork Evaluation.

	Exonum	**Quorum**	**Go-Ethereum**
- Consensus	- Custom-built BFT algorithm	- QuorumChain, IBFT and Raft-based consensus	- PoW, PoS and PoA
- Transactions p/s	- up to 5000 transactions p/s	- Dozens to hundreds	- Depends
- Private support	- Yes	- Yes	- Yes
- Smart Contract Language	- Rust	- Solidity	- Solidity
- Programming Language	- Rust	- Go, C, JavaScript	- Go, C, JavaScript
- Decentralized	- Yes	- Partially	- Optional

3.2. Blockchain Evaluation as an E-voting Service

The three blockchain systems I think in integrating and executing our smart contracts for election are comparable in Table 3. Exonum, Quorum and Geth are these.

i. Exonum - With respect to the Exonum blockchain, the full programming of the Exonum blockchain is done with Rust. For private blockchains, Exonum is created. The Byzantine algorithm is designed to achieve network consensus. Exonum allows up to 5000 transactions per second with this consensus algorithm. The drawback of the architecture is, sadly, that in the present version, Rust is the only programming language that restricts the developers to the frameworks available

in this language. In order to solve this constraint, Exonum aims to add Java-bindings and platform-independent interface specifications in the near future so that Exonum can be more developer-friendly.

ii. Quorum - Is a protocol of Ethereum-based transaction/contract privacy and new frameworks for consensus? It's a fork from Geth and it is modified according to Geth. Quorum has modified the mechanism for consensus and more concentrated on consensus algorithms in the consortium chain. It facilitates dozens to hundreds of transactions per second via this consensus.

iii. Geth - Go-Ethereum or Goth is one of three original Ethereum Protocol implementations, running smart contract apps exactly as planned without interruption, censoring, infringement or intervention from third parties [7][3]. This framework facilitates the development over and above the Geth protocol and is the most comprehensive framework for the frameworks we have evaluated. The number of transactions per second depends on whether the blockchain is deployed as a private or public network. Due to these capabilities,

Geth was the framework for this paper to accept any related blockchain platform with the same capacities as Geth.

4. DESIGN AND IMPLEMENTATION

This paper is intended to using the Icelandic identity verification service provider's electronic authentication ID Auðkenni [2], to implement a safe authentication protocol. Auðkenni uses RFID scanners from Nexus software. A user selects a PIN number consisting of 6 numbers for their corresponding ID when an electronic ID of user registers. Therefore, by scanning its ID and giving the corresponding PIN number, a user can identify himself in the voting booth.

1) Any voting district can use any computer to vote for any eligible voter because the wallet has the details for the corresponding voter in the electorate of the elector is required to vote. A correct ID and PIN number should be provided card reader and nexus apps in the electoral district for a user to successfully authenticate.

2) The related smart contract is supported for existing elections if authentication is successful. The vote is a smart contract with candidates list from which an elector may choose.

3) When an elector selects a candidate and casts his ballot, he or she signs his or her ballot by re-entering his or her electronic ID the corresponding PIN number.

4) After the vote has been signed, the voting data is checked by an appropriate district node from which the elector communicates with the smart contract. In the case that the above district node approves the voting data, the voting data of the corresponding district node must be accepted by a majority.

5) If most district nodes agree on the voting estimates, consensus has been reached on the specific vote. The user shall then obtain the transaction ID in form of a QR code and the option to print the transaction ID for with his vote the related transaction. If voting is cast and verified, the smart contract feature adds a vote to the voted party. This feature of the smart contract system is used in each voting district to decide the result of the elections. Figure 3 indicates the steps that we have just carried out.

6) All transactions obtained and checked in the current block time shall be enforced in the blockchain after the time limit for the block has been reached (see Figure 4). Each district node updates its own copy of the ledger with each new block added to the blockchain.

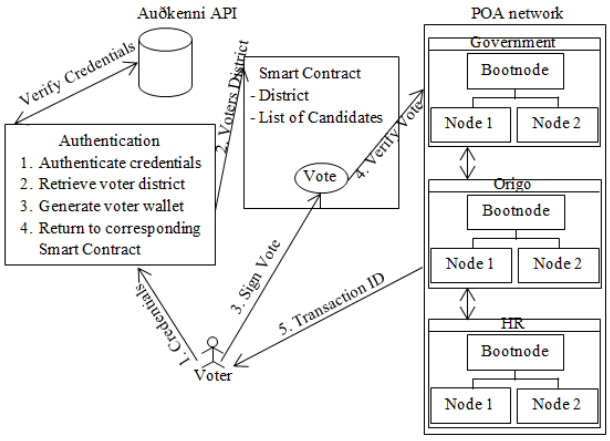

Figure 3. Voter authenticates himself and casts vote.

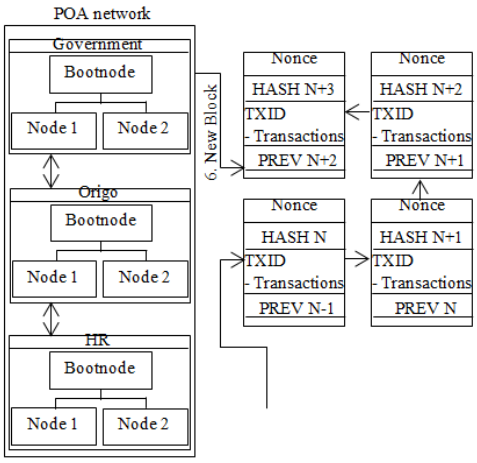

Figure 4. Block added to the blockchain.

5. CONCLUSION

It is important in modern society to change automated voting systems to make the public election process cheaper, quicker and simpler. Inexpensive and fast-tracking of the democratic process, legitimizes the voting method, eliminates a certain barrier in control between the electorate and the elected official and places certain pressure on the official chosen. Also, it creates the opportunity to a more clearly democratic system that allows electors to voice that it to particular laws and legislation. I presented a unique electronic voting system based on blockchain that uses smart contracts to allow safe and cost-effective election while securing privacy for voters in this article. I illustrated design, system security analysis and architecture of systems. In compared to the earlier studies, I have seen that blockchain technology provides new possibilities, while growing the security mechanisms of the existing scheme and new ways for transparency, for democracy countries to move from the paper-and-print election scheme to a more time-efficient election scheme. By using a

private Ethereum blockchain, hundreds of transactions can be sent to the blockchain every second, using any part of the intelligent contract to reduce the workload on the blockchain. In larger countries, some steps have to be taken to prevent increased transaction performance per second, for instance the parent and child architecture [8] that reduces a 1:100 ratio of transactions saved on the blockchain, but without affecting the security of the network. An electoral system requires individual voters to vote in their own voting district and guarantees that every single vote is counted from the right district, which may theoretically boost voting participation in the elections.

6. REFERENCES

1. Amot Fiat and Adi Shamir, "How to prove yourself: Realistic identity solutions and signature problems". https://www.math.uni-frankfurt.de/~dmst/teaching/SS2012/Vorlesung/.
2. Auðkenni, https://www.audkenni.is/en/.
3. Ethdocs.org, "What is Ethereum? - Thorium Homestead documentation", http://ethdocs.org/en/latest/.
4. Ethereum Blog, "On Public and Private Blockchains Ethereum Blog", https://blog.ethereum.org/2015/08/07/.
5. Feng Hao and Piotr Zielinsk, "A two-Round Anonymous Veto Protocol", http://homepages.cs.ncl.ac.uk/.
6. Feng Hao, P.Y.A. Ryan and Piotr Zielinski, "Two-round public dialogue on online voting", http://homepages.cs.ncl.ac.uk/.
7. Geth.ethereum.org, Go Ethereum, https://geth.ethereum.org/.
8. Jelurida, https://www.jelurida.com/sites/.
9. Konstantinos Chalkias, "Show how Zero-Knowledge Proofs works without math", https://www.linkedin.com/pulse/.
10. Manuel Blum, Alfredo De Santis, Silvio Micali and Giuseppe Persiano, "Non-Interactive Zero-Knowledge", https://people.csail.mit.edu/.
11. Mihir Bellare & Phillip Rogaway, "The Effective Protocol Architecture Framework", https://cseweb.ucsd.edu/.
12. Nicholas Weaver, "Verified today's vote", https://www.lawfareblog.com/.
13. Patrick McCorry, Siamak F. Shahandashti and Feng Hao, "A smart contract boardroom that guarantees maximum privacy for voters", https://eprint.iacr.org/.

14. Salanfe and Hacker Noon, "Setup your own private Proof-of-Authority Ethereum network with Geth", https://tinyurl.com/.
15. Sos.ca.gov, “Top-to-Bottom Review”, http://www.sos.ca.gov/elections/voting-systems/oversight/top-bottom-review/.
16. Steve Ellis, Ari Juels and Sergey Nazarov, “A Decentralized Oracle Network”, https://link.smartcontract.com/.
17. TechCrunch, “Liquid democracy uses blockchain to fix politics, and now you can vote for it”, https://techcrunch.com/.
18. The Dining Cryptographers Problem: “Unconditional Sender and Recipient Untraceability”.