The Blockchain Revolution: Paradigm Shifts in Traditional Voting Practices

Suman Mann
Assoc. Prof., Department of I.T
Maharaja Surajmal Institute of Technology
New Delhi, India

Tanya Jain
Student, Department of I.T
Maharaja Surajmal Institute of Technology
New Delhi, India

Aakash Vyas
Student, Department of I.T
Maharaja Surajmal Institute of Technology
New Delhi, India

ABSTRACT
Perhaps the most advanced application of the distributed ledger technology (DLT), the Blockchain is a decentralized system that is known to store immutable metadata with the use of robust cryptographic hashes and consensus mechanisms. The very foundation of the Blockchain is the establishment of trust-less transactions in peer-to-peer networks. Having been deemed to set off a whirlwind in Industry 4.0 as we know it and starting out with the groundbreaking Bitcoin, this relatively new technology is on the way to prove that it can find applications in almost every imaginable sector. While most people focus only on cryptocurrencies; this disruptive technology, in fact, offers utilities to many administrative operations, fintech procedures, and everyday services which could earlier only be done offline and/or in person, that can now be safely moved to the Internet as Software as a Service (SaaS) models. What makes Blockchain a powerful tool for digitalizing everyday facilities is the introduction of smart contracts, as brought forward foremost by the Ethereum platform. Considering today's technology, Blockchain may create one of the most prominent alternatives to traditional voting in terms of security, consistency and speed. The Blockchain technology, fortified by Smart Contracts, enables enhanced data verifiability and lowered costs while maintaining the openness and transparency of the voting process. The anonymity of voters, the security of ballot transmission and the veracity of votes during the billing phase are the most fundamental requirements for voting. In this paper, a potential use case of Blockchain, an E-Voting protocol, is proposed, that utilizes the Blockchain as a transparent ballot box to cast votes.

General Terms
Blockchain E-Voting Decentralized Application

Keywords
Blockchain; Voting; Ethereum; Proof of Work; Consensus; Decentralized; Democracy; Ballot

1. INTRODUCTION
Voting, whether traditional ballot based or electronic voting (E-Voting), is what modern democracies are built upon. In recent years, voter apathy has been increasing, especially among the younger computer/tech savvy generation. E-Voting is pushed as a possible solution for attracting young voters. For a vigorous E-Voting plan, various functional and security prerequisites are indicated including straightforwardness, precision, auditability, framework and information respectability, mystery/protection, accessibility, system and data secrecy/privacy[1] and availability. Blockchain, is a distributed[2] database that stores data records that continue to grow, controlled by multiple entities. The Blockchain is a trustworthy system servicing a group of nodes or non-trusting parties. Generally, Blockchain acts as a reliable third party to keep things together, mediate exchanges (transactions), and provide secure computing machines[3].

Blockchain comes with a variety of classifications, but has several common elements:

i. Blockchain is decentralized[4]; the entire recording is available for all users and peer to peer network users. This helps facilitate trust-less systems with no one central authority having monopoly of access.

ii. Blockchain uses cryptography and digital signatures to prove identity, which is the base of formation of cryptocurrencies[5] like the most popular, bitcoin[6]. Transactions can be traced back to the cryptographic identity, which is theoretically anonymous, but can be re- linked with real-life identity using reverse engineering algorithms.

iii. Blockchain has a difficult mechanism for altering stored records. Although all data can be read and new data can be written, previously existing data on Blockchain can't be changed theoretically unless the rules embedded in the protocol allow such changes. This makes Blockchains practically immutable.

iv. A Blockchain is time-stamped. Transactions in Blockchain are timed, so they are useful for tracking and verifying information. Thus, the metadata to any request for a change in node data values or states can be traced as well.

v. Blockchain is programmable. Instructions embedded in blocks, such as, "if" this "then" does that "else do this", allow transactions or other actions to be done.

2. ALGORITHMS USED

The following algorithmic concepts and technologies have served as indispensable and crucial components during the development of the various modules in the Voting Decentralized Application:

2.1 Byzantine Fault Tolerance

A distributed computing system finds itself amidst a Byzantine Fault when either a node has failed or the transmission process of information across the network faulters[7].

Blockchain systems, including the Voting System proposed in this paper, which are deemed to be Byzantine Fault Tolerant (BFT) are known to be elusive to such failures in the Information System.

2.2 Proof of Work (PoW) Consensus Mechanism Algorithm

Consensus Mechanisms are fault-tolerant algorithms in Blockchain wherein the agreement for a single data value or state in the peer-to-peer network is laid down. The Consensus Mechanism used by the Ethereum Blockchain is the Proof of Work Consensus wherein it requires the node in the Blockchain to dedicate some system resources such as processing time in exchange of deterring network abuses like denial-of-service attack, et al.[8]

2.3 Smart Contracts

A smart contract is a program or a transaction protocol which lays down the trigger events for some particular actions to happen in a Blockchain. In this Information System, the Voting Smart Contract controls the execution of specific commands in a Blockchain application according to certain agreed-upon terms and conditions, as determined by Consensus Mechanisms[9].

2.4 Merkle Tree

A Merkle Tree is a hash-based tree data structure consisting of various nodes, in which every leaf node contains the cryptographic hash of a block of data whereas every non-leaf node contains the cryptographic hash of the child nodes. In the context of the Ethereum Blockchain, Merkle Trees help in processing of large data sets for the purpose of data security and verification[10].

3. SOFTWARE DEPENDENCIES

The following software have served as indispensable and crucial components during the development of the various modules in the Voting Decentralized Application:

3.1 Ethereum

Ethereum is a decentralized open source public permissioned Blockchain which was the first ever to introduce the concept

of Smart Contracts. This Blockchain operates on Ether (ETH) which is its own cryptocurrency.

3.2 Ethereum Virtual Machine

The Ethereum Virtual Machine provides a run-time environment for Smart Contracts running on an Ethereum node. The Ethereum Virtual Machine essentially converts the Smart Contract written in solidity into machine-readable bytecode.

3.3 Ganache

Ganache is a Command Line Interface (CLI) tool that creates a local Blockchain that comes completely equipped with the features of the Main Ethereum Network, with typically ten dummy accounts (256 - bit cryptographic hash strings) loaded with virtual ETH that hold no real-world monetary value whatsoever.

3.4 Solidity

Solidity is the language of Smart Contracts, developed dedicatedly for the Ethereum Blockchain. It is much like JavaScript and also picks up some components from C#.

3.5 web3.js

web3.js is a JavaScript Library designed exclusively for Ethereum Blockchain, to create the web User Interface (UI) for Blockchain Information Systems. Web3.js is especially used for the user to be able to interact with the backend of the Blockchain application in a visual fashion.

4. LITERATURE REVIEW

The main objective of this Information System is to provide a secure and safe E-Voting[11] system or environment and show that a reliable E-Voting scheme is possible to be used on ground level using Blockchain technology. Owing to the fact that E-Voting would be available for everyone who has an internet enabled device like a computer, or mobile phone; at the very least, the opinions of people as well as some system features will be more public and accessible by the managers and politicians. This will eventually lead humanity to the true, direct and unbiased E-Democracy[12].

The very concept of E-Voting in itself is significantly older than Blockchain. All the known examples currently use means of centralized computation and different storing models. Estonia is one of the examples, since the government of Estonia was one of the first to implement a fully online directed and comprehensively reactive E-Voting solution[13].

Few other note-worthy examples include the nation of Sierra Leone, where the test was regarded as a partial deployment of a Blockchain in which the elections were only verified by Blockchain, not powered by Blockchain; and the city of Moscow's Active Citizen[14] program in 2017 wherein the organization started using a Blockchain system for voting and to make the voting results publicly available. After the voting is complete, the results were listed on a ledger that contained all the polls.

These researchers proposed a peer-to-peer Blockchain[15] based E-Voting system. They mainly focused on protection of the anonymity of the voters and their commitment to the Blockchain. They propose a very unique voting commitment format to be implemented in Blockchain. This solution has perfect base for such voting commitment format but the authors propose a different system that depends on another system that is maintained quite differently.

Since a number of devices when put together process different data collectively, new approaches emerged from this area. In this research paper, one of the hybrid models that utilizes different chains in different layers and levels which also inspired the Blockchain based E- Voting system[16] are discussed.

The first ever E-Voting system was introduced in the early 1980s by David Chaum[17]. This system used a public key cryptography, keeping the voters anonymous while casting their vote. The Blind Signature Theorem was implemented in this system so that there was no connection between the voters and ballots.

5. DESIGN PROPERTIES

The following Security Goals hold true for this E-Voting Information System which aims to be free of vulnerabilities, owing to cryptography and network security measures.

5.1 Confidentiality

This concept insinuates the prevention of data access to unauthorized users.

5.2 Integrity

This property states that all authorized users (Voters) can gain access to data or states on the Blockchain.

5.3 Availability

The Voter must enjoy complete access to the Information System.

5.4 Non-repudiation

This property ensures that the consensus established by Proof of Work remains unchallenged.

5.5 Accountability

A Blockchain Voting System is transparent and all stakeholders can monitor all activities.

6. METHODOLOGY

By virtue, a Blockchain, tracks each transaction with the help of timestamps. This renders the system to be highly transparent and, in a way, immutable. So, making changes to a transaction that has been accepted by mutual consensus becomes a nearly impossible feat. In a case of manipulation[18] of the system such as changing votes or stealing votes, other connected nodes will already be synchronized. So, the changed data will be identified instantly. Details of the system will be explained below after the use case diagram and explanation of it.

Also, the concerned Election Regulatory Authority shall determine the candidates that will be participating in that election. The ballot box information, candidates and citizen ballot box relation will be provided by the Government which is the trusted party in the elections. After citizen's vote, it is added to the Blockchain that shall be proposed below and any vote has a guarantee from the system about being immutable. Since a chain contains all the citizen votes anonymously at the end of the election, the official results will be announced within minutes after the election terminates. Any concerning outsider can get the chain and check the decisions in favor of being certain that casting a ballot is truly trusted.

This is an immense incentive for a framework that comprises of a huge number of centers and voting will happen at all the centers. For this situation, synchronization of the framework would take a lot of time. So, in order to decrease various latencies, chains are distributed over different levels. This is a huge value for a system that consists of tens of thousands of centers and there would be voting at each center simultaneously. In this case, synchronization of the system would take a lot of time. So, in order to decrease various latencies, chains are distributed over different levels.

6.1 The Architecture

There are several modules that have been employed in order to establish this Voting Decentralized Application. The Fig. 1 illustrates the architecture of the proposed Blockchain Voting Information System.

The proposed system has a leveled structure. There will be different number of levels in that system according to necessities of the country. In order to provide a fast, consistent and secure system, the module is designed in a leveled architectural manner. This number will change from country to country according to features of the country viz. demography, et cetera.

6.2 The Modularity and Working

The Voting Decentralized Application implemented on the Ethereum Blockchain has the following three modules. Here, the technicalities of accessing the project are discussed.

1. index.html – This file makes up the web page upon which the GUI of the Blockchain Voting Application has been written in HTML and CSS.
2. index.js – This is a JavaScript file that serves as the GUI (Graphical User Interface) of the Blockchain Voting Application.
3. voting.sol – This file has been written in the solidity language and lays down the smart contract for the Voting Decentralized Application (given as the block f code).

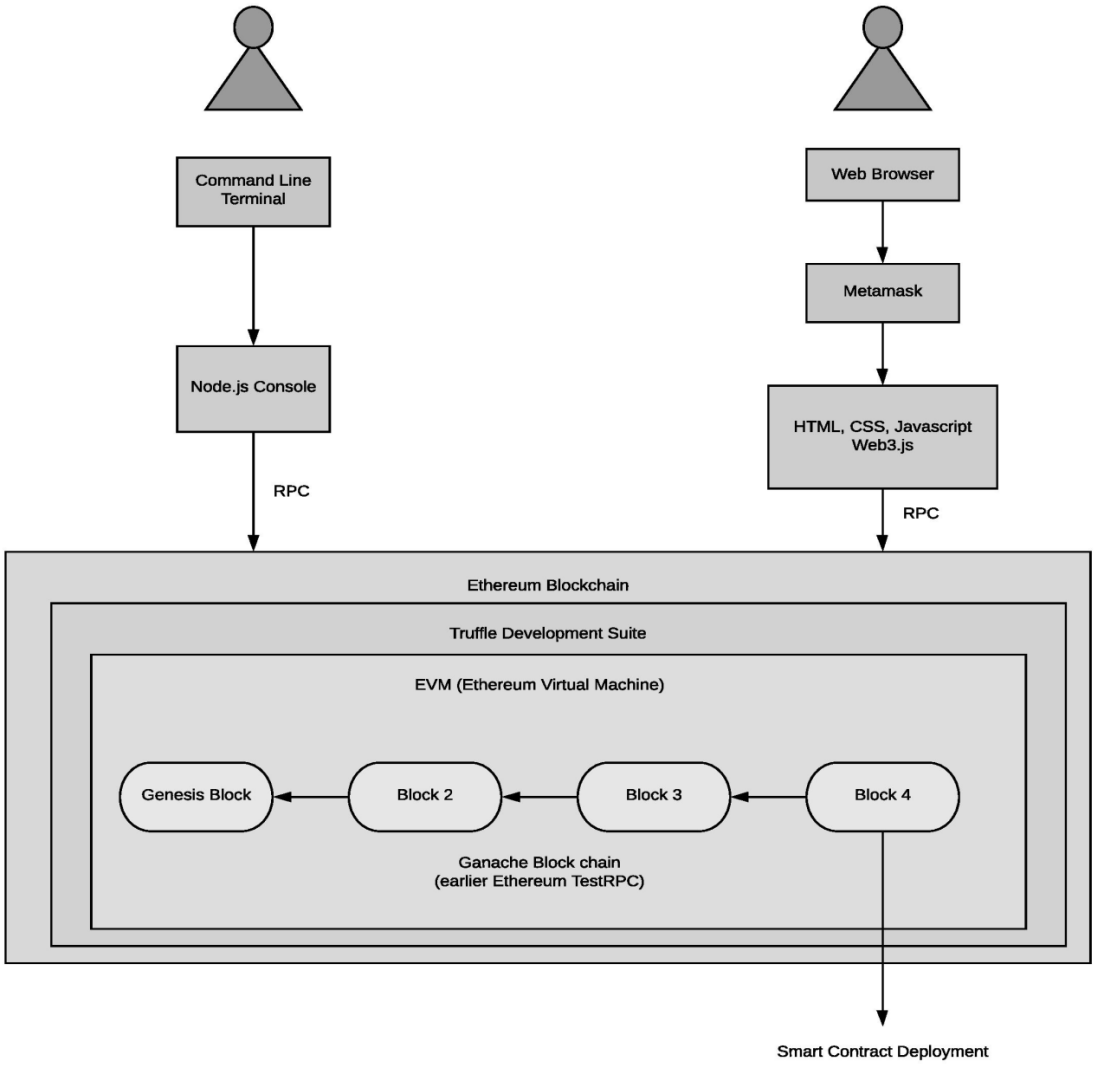

Fig 1: Architecture of Voting Decentralized Application

6.2.1 Election Smart Contract Overview

6.2.1.1 Candidate Details

The Voting Smart Contract provides the functionality of adding new Candidates that contest in an election. Here, data such as the Candidate ID, Name of the Candidate, the Vote Count for each Candidate and the Name of the Political Party that the Candidate belongs to are taken into account. As soon as the Contract is created, the constructor is triggered and it maps the aforementioned values that are mapped to the created object which is of type unsigned integer. The Smart Contract requires the Vote Count for the candidate to be zero only then that candidate can be added to the list of contestants otherwise, an error ticket is raised.

6.2.1.2 Casting Votes

Any Voter can cast a vote for any of the registered contestants by voting against the Candidate ID of the chosen individual.

The proposed system ensures that one user gets to vote just once. This is done by recording the metadata of the Ganache account of the Voter in the Blockchain. If any Voter who has previously cast a vote tries to vote again, an error ticket is raised. An error ticket is raised in the event of a Voter casting a vote for a contestant whose Candidate ID is not registered with the System, as well.

6.2.1.3 Counting Votes

Following the event of the Election, the E-Ballot is ready to post the Candidate-wise list of all the cast votes. This helps the concerned Authority to determine who stands triumphant in the Elections with the greatest number of votes.

6.3 The Command Line Access

The Information System is mainly accessed through the Command Line Interface (CLI) and below, the Linux commands.

VOTING.SOL

```
contract voting { constructor() public {} struct Candidate {

uint id; string name; uint voteCount; string party;

}

mapping(uint => Candidate) public candidates; uint public
candidatesCount;

function addCandidate(string _name, string _party) public {

require(voteTotal == 0, "Cannot submit candidate after first vote
recorded"); candidatesCount++;

candidates[candidatesCount] = Candidate(candidatesCount,
_name, 0, _party);

emit addCandidateEvent(candidatesCount);  } event
addCandidateEvent (uintindexed_candidateId); mapping(address
=> bool) public voters;

uint public voteTotal;

function vote(uint _candidateId) public {
require(!voters[msg.sender], "Vote already cast from this

address");

require(_candidateId> 0 && _candidateId<= candidatesCount,
"Candidate ID is not in range of candidates");

require(candidatesCount>= 2, "Must be at least 2 candidates
before votes can be cast");

voters[msg.sender] = true; candidates[_candidateId].voteCount++;

voteTotal++;

emit votedEvent(_candidateId);

}

event votedEvent (uint indexed_candidateId);

}
```

Fig 2: Voting Smart Contract

Step 1: First, install all dependencies mentioned in Section 2 of this paper viz. Ganache (ganache-cli), Solidity Compiler (solc) and node.js which helps in running web3.js (JavaScript Library). It is essential to have Node Package Manager (npm) installed on the system.

Step 2: Run the Ganache CLI in a Command Prompt Window

$ npm install -g ganache-cli

In a separate CLI window, execute the following command:

$ ganache-cli

Step 3: Compile the Contract using the following commands. Compile the contract from within the node.js console:

~/dapp$ node

> Web3 = require('web3

> web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));

To compile the contract, load the solidity code from voting.sol into a string variable, then compile (will run in a node.js console as well):

>code= fs.readFileSync('voting.sol').toString()

>solc = require('solc')

>compiledCode = solc.compile(code)

Step 4: Deploy the Contract by creating a contract object and deploy the contract:

>abiDefinition = JSON.parse(compiledCode.contracts[':voting'].interface)

>VotingContract = web3.eth.contract(abiDefinition)

>byteCode = compiledCode.contracts[':voting'].bytecode

>deployedContract =

VotingContract.new(['Hari','Nawaz','James'],{data:

byteCode, from: web3.eth.accounts[0], gas: 4700000})

>deployedContract.address

>contractInstance =

VotingContract.at(deployedContract.address)

Step 5: Interact with the contract

First, an interaction with the contract through a node console is established:

>contractInstance.totalVotesFor.call('Hari')

The expected output is something like this:

{ [String: '0'] s: 1, e: 0, c: [0] }

>contractInstance.voteForCandidate('Hari', {from: web3.eth.accounts[0]})

The expected output is the address of a Ganache Wallet Account which is a 256-bit string like:

'0xdedc7ae544c3dde74ab5a0b07422c5a51b5240603d31074f5b75c0ebc786bf53'

>contractInstance.voteForCandidate('James', {from: web3.eth.accounts[0]})

The expected output is the address of a Ganache Wallet Account which is a 256-bit string like:

'0x02c054d238038d68b65d55770fabfca592a5cf6590229ab91bbe7cd72da46de9'

>contractInstance.voteForCandidate('Nawaz', {from: web3.eth.accounts[0]})

The expected output is the address of a Ganache Wallet Account which is a 256-bit string like:

'0x3da069a09577514f2baaa11bc3015a16edf26aad28dffbcd126bde2e71f2b76f'$

contractInstance.totalVotesFor.call('James').toLocaleString()

The expected output is the number of votes cast: '3'

In order to use the GUI, the instance address in ~/dapp/index.js for the contract is updated.

In the node.js console, contract.Instance.address is executed, to get the address at which the contract is deployed.

contractInstance = ScoringContract.at('0xe46c0742867695226bdacc9b821d7f26dbdd294e');

~/voting-Blockchain-app/index.html can now be opened in the browser in order to view the working UI.

7. EXPERIMENTAL ANALYSIS

The Information System proposed in this paper gives an improvised solution to Naïve E-voting Systems[19]. A series of Experimental Analyses in order to compare Blockchain and Traditional Voting Systems can be drawn up in accordance with the Design Properties taken into consideration in Section 5 of this paper.

Table 1. Experimental Analysis of Comparison Between Blockchain and Traditional Voting Schema

S.No.	Parameter of Analysis	Blockchain Voting Scheme	Traditional Voting Scheme
1.	Confidentiality	The Blockchain System, as a rule, forbids the unauthorized access of data owing to public key encryption.	No advanced cryptographic techniques are typically used in a Traditional Voting System.
2.	Integrity	Integrity of data in the Blockchain System ensures that all rightful stakeholders are able to gain complete access to all data.	It often becomes possible for users to gain unauthorized access to sensitive data in Traditional Voting Systems.
3.	Availability	Availability in the Blockchain Voting System ensures ease of access to data to only those who are permitted.	Traditional Voting Systems may provide access to data that is otherwise only meant to be confidential.
S.No.	**Parameter of Analysis**	**Blockchain Voting Scheme**	**Traditional Voting Scheme**
4.	Non-repudiation	Non-repudiation in the Blockchain System ensures that the consensus of state established by Consensus Mechanisms remains unchallenged.	Owing to weak security protocols, Traditional Voting Systems often are not capable of offering indubitable validity of authorship.
5.	Accountability	There is complete transparency in this Blockchain Electoral System by virtue of the technology being put in use.	Traditional Voting Systems usually cannot vouch for robustness of algorithms.

8. CONCLUSIONS

The long-established and often primitive and/or less advanced voting practices even today find themselves victim to common fraud, dereliction of duty and authenticity. There is no presently available panacea that offers an absolute solution to all or any malpractices that are currently taking place in Electoral Systems the world over. This paper aims to provide only one such step in the Evolution of the ultimate Information System capable of uprooting all ill-complications in Electoral Systems.

This paper explores the potential of Blockchain technology and its use case in the vertical of E-Voting. Information Systems, backed by mighty cryptographic hashes especially the one suggested in the paper are able to employ Public Key Encryption for the purpose of non-repudiation.

The Blockchain Information System proposed in this paper does not claim to be averse to vulnerabilities such as malicious software attacks but as a stand-alone System; it is robust.

Based on the design of the system and the analysis carried out, it can be concluded that the system is successful functionality of recording the E-Voting System based on the Blockchain.

9. SCOPE OF FURTHER RESEARCH

This Voting Decentralized Application offers much scope for advancement. With Scientists and Engineers making quantum leaps in the relatively newly-founded field of Blockchain with every passing day, it would now be possible to integrate this project with the equally powerful domains such as Deep Learning and Artificial Neural Networks to introduce Facial Recognition and Fingerprint Scanning functionalities, IoT[19] (Internet of Things) to facilitate Remote Voting functionality, but not limited to Cyber Security to ensure this Information System against Unethical Penetrations in any manner whatsoever.

Further, improving latency and using highly efficient algorithms and newer consensus protocols make up the immediate next phases in the development of this project. The authors hope to tap the latent potential of the Blockchain technology that it offers in terms of ease of integration across various vertical and horizontal industries.

10. REFERENCES

[1] Kostal, Kristian & Bencel, Rastislav & Ries, Michal & Kotuliak, Ivan, "Blockchain E-Voting Done Right: Privacy and Transparency with Public Blockchain". 10th IEEE International Conference on Software Engineering and Service Science, At Beijing, China, 2019.

[2] Awalu, Ishaku & Kook, Park & Lim, Joa, "Development of a Distributed Blockchain eVoting System". the 10th International Conference, 2019.

[3] Zhang, Jingyu & Zhong, Siqi & Wang, Jin & Wang, Lei & Yang, Yaqiong & Wei, Boyang & Zhou, Guoyao, "A Review on Blockchain-Based Systems and Applications". International Conference on Internet of Vehicles, 2020.

[4] Priya, J & R.K., Sathia, "Disseminated and Decentred Blockchain secured Balloting: apropos to India". Tenth International Conference on Advanced Computing (ICoAC), 2018.

[5] Bystriakov, Alexandr & Andrey, Guirinskiy & Nan, Tan & Hidar, Shubbar & Din, Lin, "Crypto Currencies and Possible Risks". Digital Economy: Complexity and Variety vs. Rationality, pp.175-181, 2020.

[6] Vujicic, Dejan & Jagodic, Dijana & Ranđić, Siniša, "Blockchain technology, bitcoin, and Ethereum: A brief overview". 17th International Symposium INFOTEH-JAHORINA (INFOTEH), 2018.

[7] Kirrmann, Hubert (n.d.). "Fault Tolerant Computing in Industrial Automation"(PDF). Switzerland: ABB Research Center. p. 94. Archived from the original on 2014-03-26. Retrieved 2015-03-02.

[8] Vitalik B. (2015). Ethereum White Paper.

[9] Fries, Martin; P. Paal, Boris. "Smart Contracts" (in German). Mohr Siebeck. ISBN 978-3-16-156911-1. Retrieved 24 May 2020.

[10] Becker, Georg (2008-07-18). "Merkle Signature Schemes, Merkle Trees and Their Cryptanalysis". Ruhr-Universität Bochum. p. 16. Retrieved 2013-11-20.

[11] Curran, Kevin, "E-Voting on the Blockchain". The Journal of the British Blockchain Association, 1. 1-6. 10.31585/jbba-1-2-(3), 2018.

[12] Gurstein, Binyamin, "E-Democracy and Information Society". ICEDEG 2019 - Sixth International Conference on eDemocracy & eGovernment Quito, Ecuador, At Quito, Ecuador, 2019.

[13] F.Hao and P.Y.A. Ryan, "Real-World Electronic Voting: Design, Analysis and Deployment". CRC Press, pp.143-170, 2017.

[14] Kshetri, Nir & Voas, Jeffrey. "Blockchain-Enabled E-Voting". IEEE Software. 35. 95-99. 10.1109/MS.2018.2801546, 2018.

[15] P. Tarasov and H. Tewari, "The Future of E-Voting". IADIS International Journal on Computer Science and Information Systems, vol. 12, no. 2, pp. 148-165, 2017.

[16] Bartolucci, S., Bernat, P., & Joseph, D. "SHARVOT: Secret Share-Based Voting on the Blockchain". IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), 30-34, 2018.

[17] D. L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". Communication of the ACM. Vol. 24(2), pp. 84-90, 1981.

[18] "Final report: study on eGovernment and the reduction of administrative burden (SMART 2012/0061)". European Commission DG Communications Networks, Content & Technology, 2014.

[19] Çabuk, Umut & Adiguzel, Eylul & Karaarslan, Enis. (2018). A Survey on Feasibility and Suitability of Blockchain Techniques for the E-Voting Systems. International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE). 7. 124-134. 10.17148/IJARCCE.2018.7324.

[20] Kebede, Melkamu & Pani, Santosh, "Reshaping IOT Through Blockchain". Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019.