Research Article

Towards Secure and Auditable E-Voting System with Go Ethereum

Arya Wicaksana[1,*], Moeljono Widjaja[1],Vasaki Ponnusamy[2], M N Talib[3], Mamoona Humayun[4] ,Najm Us Sama[5]

[1]Department of Informatics, Universitas Multimedia Nusantara, Indonesia
[2]Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Malaysia
[3]Papua New Guinea University of Technology, Lae, PNG
[4]Department of Information Systems, College of Computer and Information Sciences, Jouf University, Al-Jouf, Saudi Arabia;
[5]Deanship of Common First Year, Jouf University, Sakaka 72341, Saudi Arabia

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract

The advancement of internet technology, cybersecurity, and the distributed system has enabled the development and establishment of electronic voting systems. This study focuses on the design and architecture of a secure and auditable electronic voting system that consists of a website, database system, and blockchain. The Ethereum blockchain is chosen due to its smart contract that is suitable for applications such as e-voting. The implementation is done using the Go language, and the Ethereum blockchain is accessible through the e-voting website. Initial testing and evaluation results show that the proposed design and architecture of the e-voting system allows a secure transaction to take place and auditable at the same time. Several configurations on the blockchain, such as block size, gas limit, and the number of sealers, could be modified to boost the performances further.

I.Introduction

E-voting system is getting more popular nowadays with the establishment of the internet and blockchain [1]. However, this system is faced with threats from hackers in the form of interruption, interception, modification, and fabrication. The voter may also not be confident with the integrity of the voting procedure and the confidentiality of one’s identity (anonymity). Thus, the e-voting system has to be able to provide adequate security measures while preserving user satisfaction [2]. The essential features are performing authentication, providing transparency, protecting anonymity, securing ballots, and yielding accurate statistics.

Blockchain is famous for its immutability, verifiability, and decentralized consensus peer-to-peer network [3]–[5]. Ethereum is an example of a blockchain that uses the smart contract to entrust computing tasks to the Ethereum network [6], [7]. This eliminates the possibility of downtime, censorship, fraud, or third-party interference. Related works in [1], [8]–[13] show promising progress to use blockchain technology as the foundation of the e-voting systems.

The paper’s outline is: Section II describes current works on e-voting and blockchain, Section III concludes our main contributions. Section IV presents the design of the e-voting system using Go Ethereum. Section V presents the implementation work of the e-voting system, and Section VI delivers the conclusion of the paper.

II.Related Work

Previous work [8] introduced intelligent entities that function as blockchain computing nodes that guarantee voters’ right to verify and audit the voting process. Other work, as in [9], introduced the design of a flexible and feasible e-voting design with no need for the time-triggered protocol (TTP) and fulfilled criteria on general voting systems. In the work done by Panja and Roy [10], blockchain technology is applied to the existing e-voting system. This protects verified ballots from being changed. Permissioned sidechains in [11] could also be used in the e-voting system to verify voters and record voting operations.

In [13], a set of protocols for an Ethereum based e-voting system is introduced, which allows permission checking while keeping anonymity, decoupling of checking procedure and blind signing into three steps (the organizer, the voters, and the smart contract), and the implementation based on Ethereum. It requires the next level of security and that not only steganography [14] nor cryptography [15] approaches could provide alone. Thus, blockchain could be applied to electronic voting, which is still one of the promising areas [16]. Researchers discussed the different research issues and challenges of Blockchain [17], and they used this with several applications [18,19] in different domains. The use of blockchain will make E-voting safer. Blockchain is the best possible way to enhance the security of E-Voting, it is being used in several other applications as well [20-22] to enhance the security for different domains. This will strengthen the E-voting security within required level.

III.Main Contribution

The main contributions of this work are:

a. We have introduced the architecture and protocol of an e-voting system that uses blockchain and public key infrastructure.

b. The e-voting system embodies generality, freedom, equality, secrecy, directness, and transparency.

c. The web-based e-voting system that is built using JSON-RPC over HTTP with WebSocket and IPC powered by Ethereum.

IV.Design

The complete client-server architecture of the e-voting system is presented in Fig.1. The e-voting system consists of a web server, database server (MySQL), and the Ethereum network. The web server hosts the website for public access to the Ethereum network. The database server stores all information regarding the blockchain-related transactions generated by the website and also the information required for the content management system. The Ethereum network is the place where all transactions are safely stored in the blockchain.

The users could access the e-voting system through the website and the orange person node. The orange person node in Fig.1 is the direct entry point to the Ethereum network for public uses. This node could run as either as a full node (by default), or as an archive node (to retrieve and retain all data, including status and history), or as a light node (to retrieve the latest live data rapidly). The primary purpose is solely to allow legitimate third party institutions to obtain the true copy of the blockchain for auditing purposes (transparency). Other processes could also access this node through the internet using either JSON-RPC (remote procedure call) over HTTP, WebSocket, or inter-process communication (IPC) transport. Thus, illegitimate agents are forbidden to access the Ethereum network directly through this node.

The consensus algorithm that is used in the Ethereum network is the Clique algorithm that is proof-of-authority (PoA). In PoA, the consensus is reached with a majority (51%) agreement among validators (miners). The default number of validators of the e-voting system is three (2 miners and one non-miner) with the possibility to add more validators on-the-fly. This allows the e-voting system owner to decide whether or not the mining process exclusively belongs to the owner or to be shared with others.

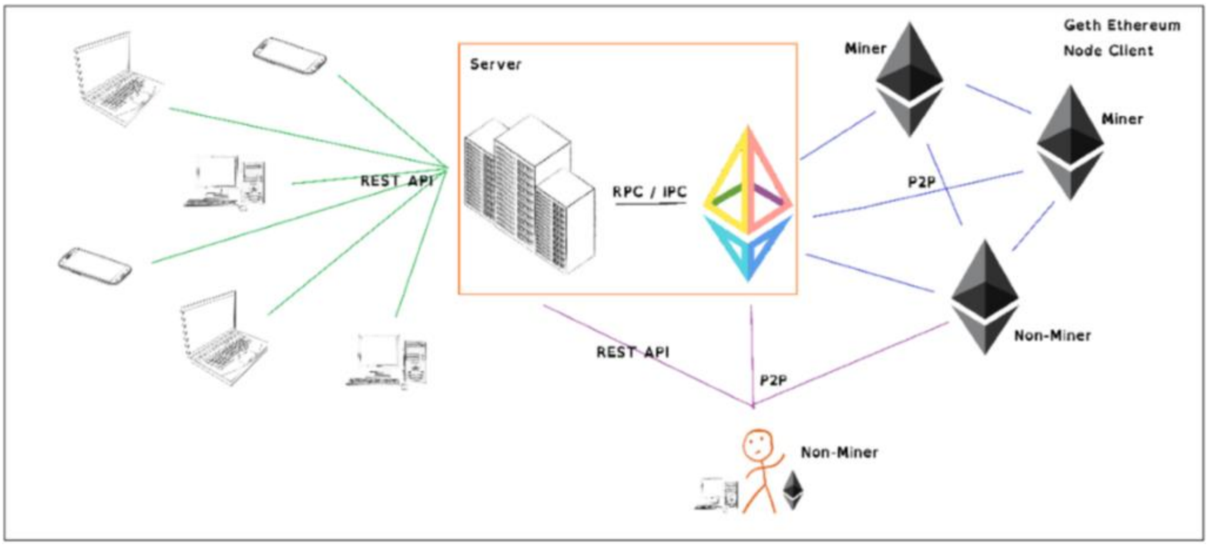


Fig.1 The proposed e-voting system architecture based on Go Ethereum

The web server is connected to an Ethereum node exclusively and privately to ensure maximum security on the communication channel. Every transaction that is generated from the website is logged into the MySQL database to allow auditing purposes. The database consists of seven tables: candidates, contracts, elections, funds, participants, transactions, and users. Table transactions record all transactions generated from the website to the Ethereum network. This allows the e-voting system to be audited in two ways: all transactions generated by the website and all transactions stored in the blockchain in the Ethereum network.

The e-voting system maintains two roles for the users: voter and organizer. The organizer could set up an election in the e-voting system, including registering and validating the voters. The voting process could only be done through the website. Public users that have permission to connect to the Ethereum network directly are not able to vote. This voting process is done by sending a transaction to the smart contract (secrecy). The e-voting system could hold several elections simultaneously using the same Ethereum network (generality). Hence, voters are also allowed to join more than one election.

The users are provided with an Ethereum wallet that contains 1 ETH (ether) coin by default. Only the owner of the e-voting system could give more ether to the users. This guarantees the equality aspect of the e-voting system. This ether serves as the ballot for the election. The Ethereum network consumes the gas price of the voting process carried out by the smart contract. The voting process is done on the website by choosing the candidate in the election (directness). Voters could later see their selection but could no longer modify it.

The Ethereum network is designed to be a private and trusted network. This is achieved by the configuration of the genesis block of the blockchain. In this work, the difficulty to mine the block is set to the lowest value possible. Functionalities that are served by the smart contract are the followings:

1. vote (electionId, candidateAddress, voterAddress)
2. showMyVote (electionId, voterAddress)
3. showResultCount (electionId, candidateAddress)
4. createElection (electionId)
5. addCandidate (electionId, candidateAddress[])
6. createElectionWithCandidates (electionId, candidateAddress[])
7. addParticipant (electionId, voterAddress[])
8. endElection (electionId)

V.Implementation

Fig.2 presents the homepage of the website. On the website, users must log in to perform any activities related to voting. The website also features the block explorer page that is accessible by the public. The website is hosted at Heroku, and it is connected to an Ethereum network hosted in the local machine.

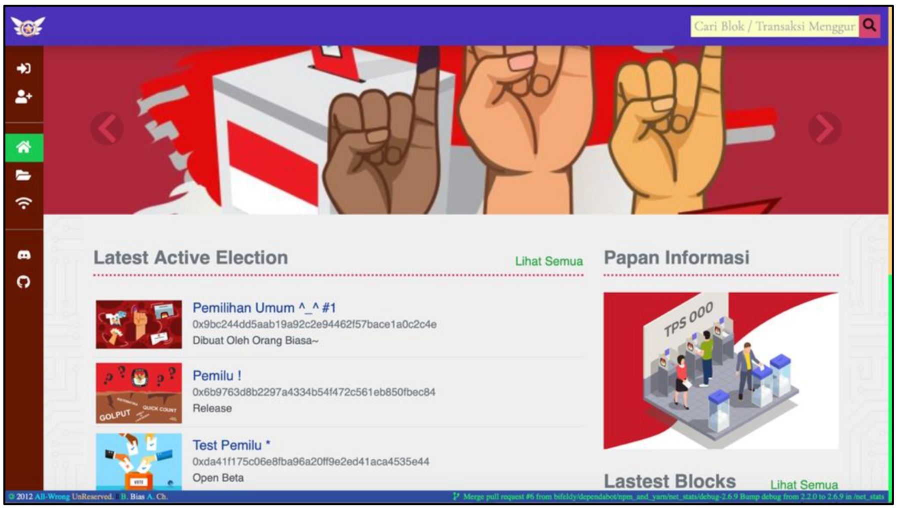
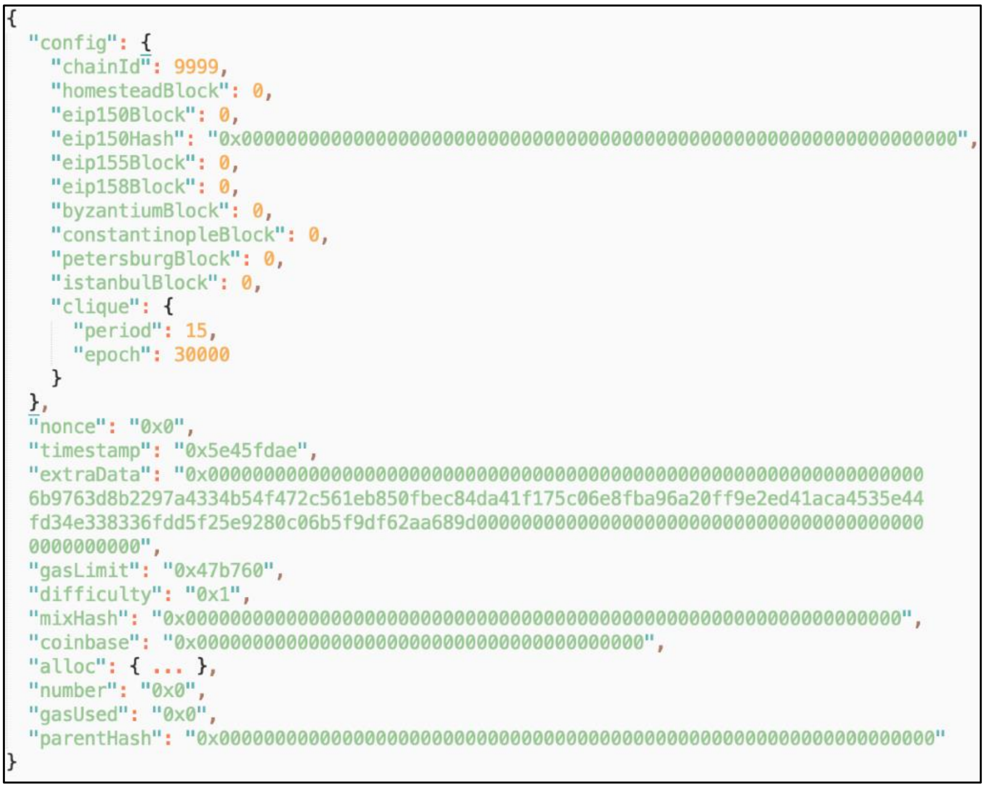

Cari Blok / Transaksi Menggur
Latest Active Election
Lihat Semua
Pemilihan Umum ^_^ #1
0x9bc244dd5aab19a92c2e94462f57bace1a0c2c4e
Dibuat Oleh Orang Biasa~
Pemilu !
0x6b9763d8b2297a4334b54f472c561eb850fbec84
Release
Test Pemilu *
0xda41f175c06e8fba96a20ff9e2ed41aca4535e44
Open Beta
Papan Informasi
TPS 000
Lastest Blocks
Lihat Semua
© 2012 All-Wrong UnReserved. B. Bias A. Ch.
Merge pull request #6 from bifeldy/dependabot/npm_and_yarn/net_stats/debug-2.6.9 Bump debug from 2.2.0 to 2.6.9 in /net_stats

Fig.2 The e-voting system

A. Genesis Block

The blockchain id is set to 9999 in the genesis block. The configuration of the genesis block is shown in Fig.3. The clique period is set to 15, and the epoch is set to 30,000. The difficulty is set to the lowest value possible (0x1). Also the variable gasUsed is set to 0.

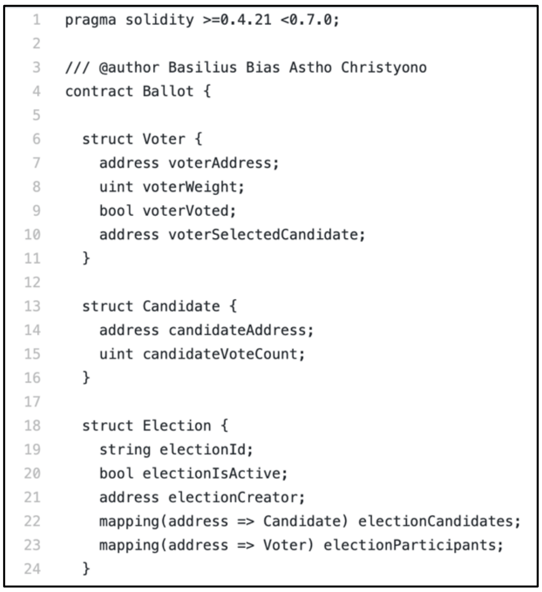


```
{
  "config": {
    "chainId": 9999,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "clique": {
      "period": 15,
      "epoch": 30000
    }
  },
  "nonce": "0x0",
  "timestamp": "0x5e45fdae",
  "extraData": "0x0000000000000000000000000000000000000000000000000000000000000000
  6b9763d8b2297a4334b54f472c561eb850fbec84da41f175c06e8fba96a20ff9e2ed41aca4535e44
  fd34e338336fdd5f25e9280c06b5f9df62aa689d0000000000000000000000000000000000000000
  0000000000",
  "gasLimit": "0x47b760",
  "difficulty": "0x1",
  "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "coinbase": "0x0000000000000000000000000000000000000000",
  "alloc": { ... },
  "number": "0x0",
  "gasUsed": "0x0",
  "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000"
}
```

Fig.3 Genesis block configuration

B. Smart Contract

The smart contract is utilized here to define the functionality of the blockchain. Fig.4 presents part of the implementation of the smart contract. The variable voterAddress is the public address of the voter. The variable voterWeight defines the value (the weight) of the vote. Variable voterVoted is a flag to mark voters who have already voted. The voterSelectedCandidate holds the public address of the candidate. Struct Candidate keeps the public address of the candidate and the number of votes received by the candidate. Struct Election keeps the election id, the status of the election (electionIsActive), the public address of the organizer (electionCreator), list of candidates' public address (electionCandidates), and list of voters' public address (electionParticipants). The rest of the smart contract implementation describes the functionalities carried out by the ballot as specified in Section III.



```
pragma solidity >=0.4.21 <0.7.0;

/// @author Basilius Bias Astho Christyono
contract Ballot {

  struct Voter {
    address voterAddress;
    uint voterWeight;
    bool voterVoted;
    address voterSelectedCandidate;
  }

  struct Candidate {
    address candidateAddress;
    uint candidateVoteCount;
  }

  struct Election {
    string electionId;
    bool electionIsActive;
    address electionCreator;
    mapping(address => Candidate) electionCandidates;
    mapping(address => Voter) electionParticipants;
  }
```

Fig.4 Part of implementation of the smart contract

Information regarding voting-related transactions stored in the database system are transaction id, blockHash, blockNumber, contractAddress, cumulativeGasUsed, gasUsed, and transactions.from (sender public address), transactions.to (receiver public address), logsBloom, status, transactionHash, transactionIndex, and createdAt.

C. Testing and Evaluation

The preliminary testing results show that some configurations could affect Ethereum network performance. This test's Ethereum network consists of three nodes: 2 miners running in local virtual machines with a different public IP address and one non-miner running in the AWS virtual machine. The main configurations that contributed to it the most are the number of validators, the minimum block time (clique. Period), and the gas limit. Another is the number of validators in the Ethereum network. The increase in the number of validators will also increase the latency in the Ethereum network. This could lead to synchronization problems during the creation of a block.

Each configuration point is tested by running the blockchain network for at least an hour. The propagation block time is influenced mainly by many validators (sealers) and the time needed to make new blocks. The results are shown in Fig.5 and Fig.6 for the gas limit of four million wei and eight million wei, respectively.

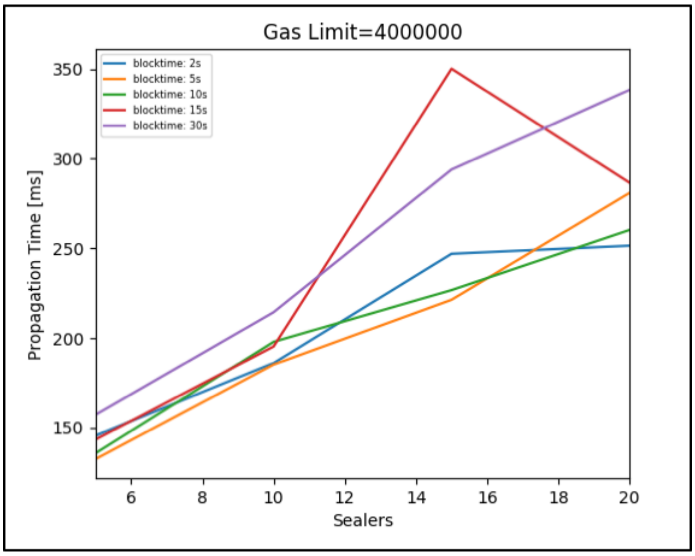


Fig. 5 Block propagation with gas limit of 4 million wei

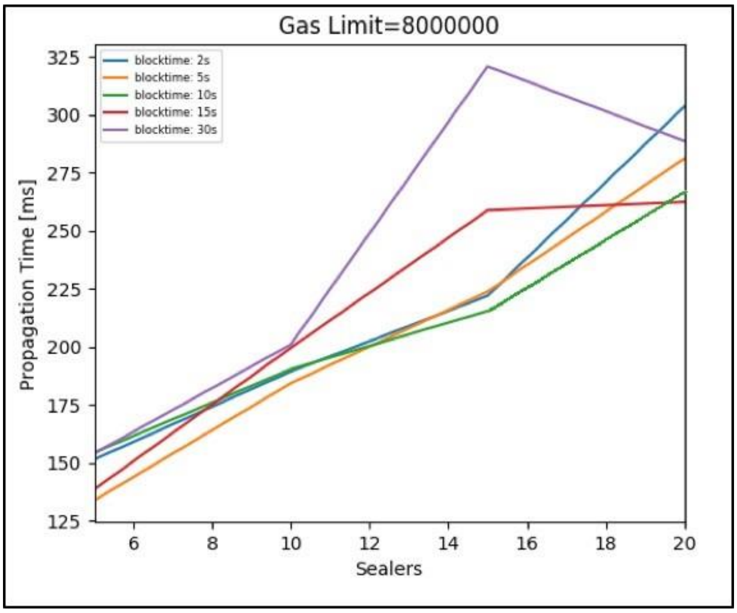


Fig. 6 Block propagation with gas limit of 8 million wei

Based on Fig.5 and Fig.6, it is observable that the propagation time is dependent on the number of sealers. Higher block times also affect the delays, which is explainable because higher block times result in larger block sizes. Thus, this takes more time due to the larger size of data to be transmitted. Delays also affect the consensus process that could lead to lost blocks, as presented in Fig.7.

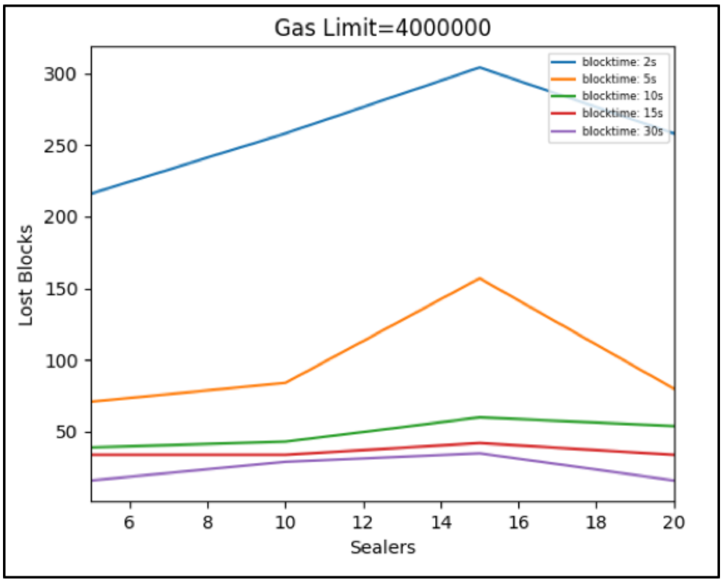


Fig. 7 Number of lost blocks

VI.Conclusion

This study produced a design and architecture of electronic voting system that consists of website, database, and blockchain. The Go Ethereum blockchain delivers main functionalities for the voting process in the e-voting system. The system architecture is built by combining the client-server of the web and database server with the peer-to-peer network of the blockchain. The Ethereum network used in this work is configured to be private and trusted. The architecture allows the e-voting system to be audited securely and transparently from the outside by connecting to the given public node. Testing shows that the genesis block configurations affects the performances of the blockchain including the PoA algorithm and the number of sealers used. Further studies are required to find the optimum configuration of the network to enhance the e-voting system's performance. This includes an evaluation of the security aspects and functional requirements of the e-voting system.

REFERENCES

[1] Y. Lee and D. Won, “A practical and secure electronic election system,” *ETRI J.*, 2012, doi: 10.4218/etrij.12.0111.0121.

[2] D. A. Gritzalis, “Principles and requirements for a secure e-voting system,” *Computers and Security*. 2002, doi: 10.1016/S0167-4048(02)01014-3.

[3] M. Humayun, N. Jhanjhi, B. Hamid and G. Ahmed, "Emerging Smart Logistics and Transportation Using IoT and Blockchain," in IEEE Internet of Things Magazine, vol. 3, no. 2, pp. 58-62, June 2020, doi: 10.1109/IOTM.0001.1900097..

[4] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A survey on the security of blockchain systems,” *Futur. Gener. Comput. Syst.*, 2020, doi: 10.1016/j.future.2017.08.020.

[5] A. P. Singh et al., "A Novel Patient-Centric Architectural Framework for Blockchain-Enabled Healthcare Applications," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2020.3037889..

[6] G. Wood, “Ethereum: a secure decentralised generalised transaction ledger,” *Ethereum Proj. Yellow Pap.*, 2014.

[7] V. Buterin, “Ethereum White Paper,” *Etherum*, 2014.

[8] M. Pawlak, A. Poniszewska-Maránda, and N. Kryvinska, “Towards the intelligent agents for blockchain e-voting system,” 2018, doi: 10.1016/j.procs.2018.10.177.

[9] Y. Liu and Q. Wang, “An E-voting Protocol Based on Blockchain,” *IACR Cryptol. ePrint Arch.*, 2017.

[10] S. Panja and B. K. Roy, “A secure end-to-end verifiable e-voting system using zero knowledge based blockchain,” *IACR Cryptol. ePrint Arch.*, 2018.

[11] F. Fusco, M. I. Lunesu, F. E. Pani, and A. Pinna, “Crypto-voting, a blockchain based e-voting system,” 2018, doi: 10.5220/0006962102230227.

[12] H. Yi, “Securing e-voting based on blockchain in P2P network,” *Eurasip J. Wirel. Commun. Netw.*, 2019, doi: 10.1186/s13638-019-1473-6.

[13] Q. Zhang, B. Xu, H. Jing, and Z. Zheng, “Ques-chain: An ethereum based e-voting system,” *arXiv*. 2019, doi: 10.5121/csit.2019.90803.

[14] N. Sofian, A. Wicaksana, and S. Hansun, “LSB steganography and AES encryption for multiple PDF documents,” 2019, doi: 10.1109/CONMEDIA46929.2019.8981842.

[15] C. D. Budianto, A. Wicaksana, and S. Hansun, “Elliptic Curve Cryptography and LSB Steganography

for Securing Identity Data," 2020, doi: 10.1007/978-3-030-25217-5_9.

[16] D. Di Francesco Maesa and P. Mori, "Blockchain 3.0 applications survey," *J. Parallel Distrib. Comput.*, 2020, doi: 10.1016/j.jpdc.2019.12.019.

[17] Alamri, M., Jhanjhi, N. Z., & Humayun, M. (2019). Blockchain for Internet of Things (IoT) Research Issues Challenges & Future Directions: A Review. Int. J. Comput. Sci. Netw. Secur, 19, 244-258.

[18] A. P. Singh et al., "A Novel Patient-Centric Architectural Framework for Blockchain-Enabled Healthcare Applications," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2020.3037889.

[19] Amir Latif, R.M., Hussain, K., Jhanjhi, N.Z. et al. A remix IDE: smart contract-based framework for the healthcare sector by using Blockchain technology. Multimed Tools Appl (2020). https://doi.org/10.1007/s11042-020-10087-1

[20] M. Humayun, N. Jhanjhi, B. Hamid and G. Ahmed, "Emerging Smart Logistics and Transportation Using IoT and Blockchain," in IEEE Internet of Things Magazine, vol. 3, no. 2, pp. 58-62, June 2020, doi: 10.1109/IOTM.0001.1900097.

[21] A. Ullah, M. Azeem, H. Ashraf, A. A. Alaboudi, M. Humayun and N. Jhanjhi, "Secure Healthcare Data Aggregation and Transmission in IoT—A Survey," in IEEE Access, vol. 9, pp. 16849-16865, 2021, doi: 10.1109/ACCESS.2021.3052850.

[22] E. Golden Julie, J. Jesu Vedha Nayahi and Noor Zaman Jhanjhi, Blockchain Technology: Fundamentals Applications and Case Studies[M], 09 2020.