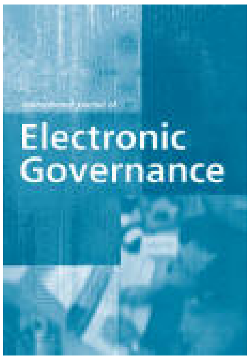


International Journal of Electronic Governance

ISSN online: 1742-7517 - ISSN print: 1742-7509
https://www.inderscience.com/ijeg

Blockchain and smart contract enabled smart and secure electronic voting system

Kailash Chandra Bandhu, Ratnesh Litoriya, Murtaza Bagwala, Alakmar Barwaniwala, Manas Garg

DOI: 10.1504/IJEG.2023.10054587

Article History:

Received:	22 March 2022
Last revised:	07 May 2022
Accepted:	19 June 2022
Published online:	05 April 2023

Blockchain and smart contract enabled smart and secure electronic voting system

Kailash Chandra Bandhu*, Ratnesh Litoriya, Murtaza Bagwala, Alakmar Barwaniwala and Manas Garg

Department of Computer Science and Engineering,
Medi-Caps University,
Indore, 453331, India
Email: kailashchandra.bandhu@gmail.com
Email: litoriya.ratnesh@gmail.com
Email: murtazabag786@gmail.com
Email: alakmarbarwaniwala@gmail.com
Email: manasgarg1999@gmail.com
*Corresponding author

Abstract: Building an electronic voting system is a prime requirement of a government election system to avoid the manipulation in the voting machines and make them more secure and efficient. There is a need of cost-efficient, time-saving, and trusted voting system. Blockchain technology is a revolutionary method that provides decentralised, distributed, and immutable ledgers features. The proposed work utilised this recent technology along with smart contract and keccak256 encryption algorithm to implement vote casting. The performance of the system is measured based on the execution time of the smart contract, average voting time per user, and the hidden and visible gas cost for smart contract deployment for voting. The results obtained come to be promising with an average execution time of 6ms per vote and the percentage of visible cost out of voting and contract deployment is 15.42% and hidden costs out of voting and contract deployment is 84.58%.

Keywords: electronic voting; blockchain; smart contracts; ethereum; distributed computing.

Reference to this paper should be made as follows: Bandhu, K.C., Litoriya, R., Bagwala, M., Barwaniwala, A. and Garg, M. (2023) 'Blockchain and smart contract enabled smart and secure electronic voting system', *Int. J. Electronic Governance*, Vol. 15, No. 1, pp.56–73.

Biographical notes: Kailash Chandra Bandhu received his BE, MTech and PhD degree in Computer Science and Engineering from reputed Universities of India in 2005, 2010 and, 2017 respectively. He is an Associate Professor in the Department of Computer Science and Engineering, Medi-Caps University, Indore, Madhya Pradesh, India. His research interest includes machine learning, big data analysis, wireless network and blockchain technology. He supervised various BTech. Projects on Machine Learning, Blockchain Technology and Internet of Things. He also supervised MTech and PhD research scholars.

Ratnesh Litoriya received his BTech (Information Technology), ME (Computer Engineering), and PhD (Computer Engineering) degrees from different reputed

Universities of India in 2004, 2007, and 2015, respectively. He has been with the Department of Computer Science and Engineering, Medi-Caps University Indore, India, where he is currently an Associate Professor. His research interests covers software engineering, machine learning, Fuzzy intelligence, elderly care, Blockchain technology, and their application areas. He is a Microsoft certified professional in dot net technology and the recipient of International Award for Professor with Huge Potential in Engineering conferred by World Federation of Science and Technology. He has published various research papers in international journals of repute. He has also published an Indian patent for intelligent and adaptive control for micro hydro plant. He has been on the Editorial Board of several International journals.

Murtaza Bagwala persuing BTech in Computer Science and Engineering, Medi-Caps University, Indore, India. His specialisation in blockchain, artificial intelligence and machine learning.

Alakmar Barwaniwala persuing BTech in Computer Science and Engineering, Medi-Caps University, Indore, India. His specialisation in blockchain, artificial intelligence and machine learning.

Manas Garg persuing BTech in Computer Science and Engineering, Medi-Caps University, Indore, India. His specialisation in blockchain, artificial intelligence and machine learning.

1 Introduction

In any democratic country voting is a very important aspect of choosing the government, as it decides the fate of the country for the next few years. That's why implementing a good voting system is very necessary. The current voting system is fit for a large amount of population but it has its flaws. The voting is crucial for all democratic society, and every eligible citizen of the nation or organisation can take part. In the current voting system, voting stations placed in certain areas often create trouble for voters due to long queues.

At the top of the whole democratic framework, building trust is the ultimate goal of electoral reform by implementing a secure and transparent e-voting solution. Any credible electoral process that enjoys widespread public trust and confidence should be supported. Voter trust is based mostly on the social and political environment in which any e-voting solution is put into use. issues in this context can be directly addressed by a comprehensive, reliable, and financially viable blockchain and smart contract-based e-voting strategy, while others, such as a general lack of trust in the electoral management body (EMB) or fundamental political or technical opposition, will be more difficult to change. This is where the E-Voting system comes to play as an alternative to the traditional voting system. Creating an Online voting system is easy to implement but it has its own set of drawbacks just as a traditional system which is filled with loopholes. The proposed solution acts differently from any generic online solution as it uses the concept of blockchain and smart contracts to build the more secure application. It provides flexibility to the voters, so that authenticated voters can vote from anywhere.

It also provides the voter confidentiality, security, transparency features in the voting system.

The security community has viewed traditional voting machines as flawed because there is always potential that anybody with physical access to traditional machines can manipulate it or manipulate the votes which would make a mockery of the election system.

A blockchain is a public ledger that is distributed, immutable, and indisputable. The four primary elements of this innovative technology are as follows (Hjalmarsson et al., 2018; Kabra et al., 2020; Pandey and Litoriya, 2020b):

i The ledger can be found in a variety of places: There is no single point of failure in the distributed ledger's upkeep.

ii To add any new block in the decentralised ledger of blockchain, blockchain refers to a previous version of the ledger, which forms an immutable chain that doesn't allow manipulation with previous entries' integrity, that's why it is called 'blockchain'.

iii To add any entries of a new block to and to make them a permanent part of the ledger, a majority of the nodes present in the network must give their permission to allow it. These technological elements use modern encryption to provide a level of security equal to or greater than any other database. Many people, including ourselves, believe that blockchain technology is the right tool for developing a new modern democratic voting procedure.

One of the main points of focus in any election during the election period is about security and whether the votes are secure or not, another thing focuses on is that an eligible member of the voting community or organisation can vote only once, else it would breach the rules of the election and there would be no point in conducting such election where one individual can vote more than once. The proposed system uses smart contracts to create a complex data type for each candidate that will participate in the election. Each candidate has a candidate ID, candidate name, and candidate vote attributes. When a user vote using the frontend, it increases the vote count for that particular candidate using a smart contract. Smart contracts are programs that execute when certain predefined criteria are satisfied and are maintained on a blockchain. i.e., mention conditions on smart contract, and when those are met, it follows whatever agreement is written in the smart contract. Smart contracts are used to complete the execution of an agreement without any manual help i.e., automatically, so that everyone involved in the smart contract knows the outcome depending on which conditions are met, it also saves a lot of time (Buterin, 2014; Gupta et al., 2020). This work automates the workflow and triggers the action if the conditions are satisfied. The Benefits of using smart contract are (Gupta et al., 2020; Soner et al., 2021):

1 *Autonomous*

The smart contract provides many good features, one of them is autonomy. In simple terms, it means that there will be no interruptions, and no third parties will be able to change the agreement or decision. This automation can go a long way in assisting businesses in automating some areas of their operations. Not only that, but it also resolves trust concerns in various processes.

2 *Secured*

Smart contracts' security is another feature that sets them apart. It allows processes to operate safely. Smart contracts also function properly as a result of the encryption. The data generated by smart contracts cannot be updated or altered in any manner since they function on networks with immutable data, and it ensures that all the data are secure.

3 *Interruption free*

Smart contracts do not have any downtime. This implies they cannot be avoided or interrupted once they've started running.

4 *Trust-less*

The system as a whole is untrustworthy. This eliminates the need to rely on other parties. Doesn't it seem counter-intuitive? To put it another way, it simply implies that don't need the trust from the parties for a transaction to complete. A transaction or a deal does not necessitate trust as a component. Because smart contracts run on a decentralised network, the entire network is trustworthy.

5 *Cost-effective*

Transactions are more cost-effective using smart contracts. This is achieved by reducing intermediates from the process. These speeds up transactions while also removing the costs connected with them.

6 *Fast performance*

Compared to the old-fashioned traditional approach, autonomous smart contracts performed far faster. Because all of the parameters have already been set in the smart contracts, it merely needs to match them before it can begin to execute.

This paper aims to propose blockchain technology-based implementation for secure vote casting using smart contract and keccak256 encryption algorithm.

2 Literature survey

The problem of making vote casting is addressed by several researchers and practitioners. The cost involves in the conduction of elections frequently in any democratic country is a significant concern to promote such type of research. This section highlights the notable contribution in this field also discusses the limitations and research gaps that may be worked upon. The applications of Blockchain technology may be find in many different areas like Health care, Banking finance, industrial automation, Education, Supply chain, Accident prevention, Agriculture, land record management, Software engineering etc. (Guo and Liang, 2016; Wijaya et al., 2017; Grech and Camilleri, 2017; Cai et al., 2018; Mengelkamp et al., 2018; Bodkhe et al., 2019; Leeming et al., 2019; Pandey and Litoriya, 2021; Dubey et al., 2020; Pandey and Litoriya, 2020a, 2020c, 2020d; Qashlan et al., 2021; Soner et al., 2021, 2022; Verma et al., 2021). The possibility of applying this technology is also been explored for electronic voting few incremental works are summarised below.

In recent years, blockchains and smart contracts have gotten a lot of attention in a variety of fields (Khan et al., 2021). Smart contracts on the blockchain are ordinary agreements created in computer programs that encode an understanding between untrustworthy parties (Alharby et al., 2018). If certain circumstances are met, smart contracts are performed on a blockchain system, eliminating the need for a trusted third party (Agrawal et al., 2022). The blockchain can host a new smart contract by invoking the function Object() [native code] via a transaction whose sender becomes the smart contract owner. Another function that may be defined in a smart contract is the self-destruct function. In most cases, only the smart contract owner can use this function to terminate the contract. Adoption of Smart contracts in many areas along with e-voting received encouraging results (Al-madani et al., 2020).

Yi (2019) presents the techniques for securing e-voting based on blockchain in peer-to-peer networks and synchronised model was designed to avoid the counterfeiting of votes which is based on distributed ledger. The elliptic curve cryptography (ECC) algorithm was used to designed this model for user authentication and non-repudiation. The implementation of the system was done on the Linux platform for multiple candidates and the python programming was used as a source code. There is no involvement of third party and real identity of the users, only SHA256 were used for generation of the system id corresponding to each user of the system.

Shejwal et al. (2019) used visual cryptology method of 2 out of 2 schemes which is based on predetermined turn on the system on blockchain and this is different from bitcoin which uses proof of work concept. This new Proof of vote system had at least Nc/2+1 commissioners working on it which guaranteed the security of the transaction.

Agbesi (2020) conducted a study which examined and understood factors that influence internet voting (i-voting) adoption intention from young voters' perspective, with the help of unified theory of acceptance and use of technology (UTAUT).

Jaiswal et al. (2021) proposed an idea which was based on the AES256 encryption algorithm for verification of the user. AES256 is a symmetric key cipher. The 32-bit secret key given to the user during registration is used to confirm the voter legitimacy. LevelDB was used as the database to record and cast the vote. LevelDB database provides the immutability features as used by the bitcoin. This ensures that anyone with access to the system cannot modify the votes and manipulate the result. Firebase authentication was used as the user authentication API.

Jafar et al. (2021) reviewed blockchain for electronic voting systems and discussed open research challenges. This work compared various blockchain-based electronic voting platforms. These various online voting platforms were evaluated based on security characteristics such as audit, privacy, voter falsifiability, authenticity, accessibility, scalability, precision, and affordability. Scalability analysis of various blockchain frameworks was also discussed. The blockchain frameworks were compared on the basis of block generation time, hash rate, rate of transactions, the cryptographic technique used, power consumption, and scalability.

Lai et al. (2018) proposed a system which provided a low-level trust between the parties is known as Decentralised Anonymous Transparent Electronic Voting System. According to them the existing voting mechanism is more suitable for large-scale electronic elections. Unfortunately, because third-party was not involved, after the election process the scheme is accountable for audit the vote, their proposed solution is not robust enough to protect against DoS assaults. Because of the platform's limitations, this solution is only viable for modest sizes. Although ring signature protects individual voters' privacy, it is difficult to organise and coordinate multiple signer entities. This work used the proof of work consensus mechanism, it had significant downsides like increase in energy consumption due to miners' supercomputers which computes millions of calculations per second. This setup is costly and energy-intensive because it necessitates a lot of computing power.

The reliable electronic voting mechanism Basit Shahzad and Jon Crowcroft (BSJC) proof of completeness was suggested by the (Shahzad and Crowcroft, 2019). This work used a process model for describing the overall structure of the electronics voting system. This work focused on anonymity, privacy, and security issues of small-scale election. However, a few other issues have raised. It uses proof of labor which has complex mathematical calculations and it consumes lot of time. As it involves a third party, it affects end-to-end verification due to the data tampering, leakage, and unfair results. The polling procedure may be delayed if the block is generated and sealed on a big scale.

A blockchain-based anti-quantum electronic voting machine with an audit feature has been proposed by Gao et al. (2019). They modified the code to make it more secure against quantum attacks in the Niederreiter algorithm. It protects the voter's confidentiality using key generation center which is certificate less cryptosystem, it also makes the audit process easier. An analysis of their approach, the security and efficiency gains are significant for limited number of voters for a small-scale election. If the number is large, some efficiency is traded to improve security.

Khan et al. (2020) proposed a system which was used to test permission and permissionless architecture. It was tested under different conditions such as size, generation speed and transaction rate of block and voting population known as Block-based E-voting Architecture. After analyzing the test performed it was found that these factors affected the scalability and efficiency of the digital voting system, including parameters as security and performance. Generation of voter and candidate addresses are the main agenda of their electoral procedure. Candidates were assigned votes using the addresses that voter had. A miner group is created for maintaining the ledger of blockchain and to keep the number of votes secure. Main ledger is updated once miners update the status of voting. However, there are certain faults in this paradigm. There is no mechanism to prevent ineligible voters from voting, and the system is vulnerable to quantum attacks. They employed the multichain framework, which is a private blockchain developed from bitcoin that is inappropriate for nationwide voting. As stated by the authors, this approach is only ideal for small and medium-sized voting contexts.

After doing the literature research survey, it is identified that the various researchers worked on the e-voting system using multichain framework, visual cryptography, Key Generation Center (KGC), Distributed Ledger Technology (DLT), ECC in the blockchain. The problem is solved using firebase authentication and by creating and mining the blocks. The problem was not solved by using other platforms like ganache, truffle, solidity, web 3.0, and metamask and still there is scope to continue work on it.

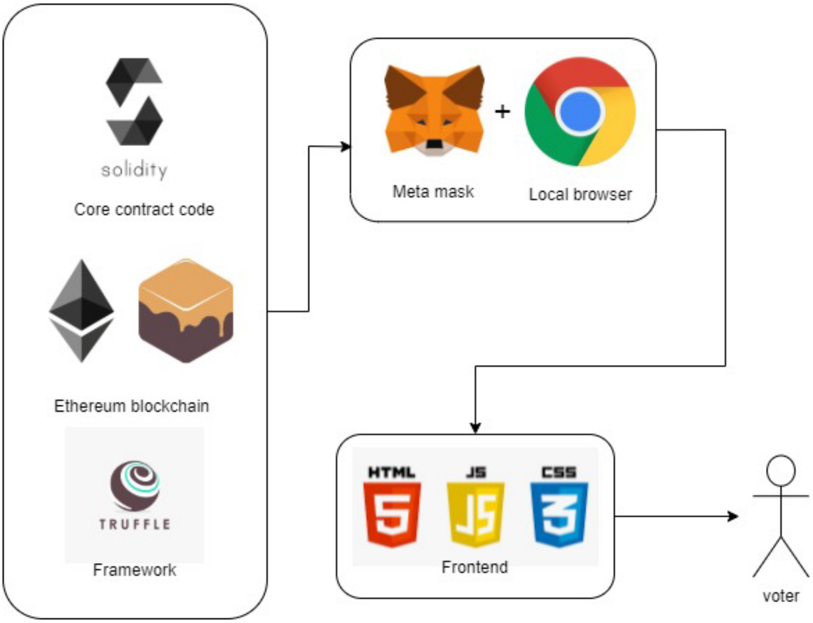
3 Proposed methodology

The research method applied in this paper is theoretical in nature, using modeling to simulate a digitalised and remote electronic voting environment. The blockchain is a network and a database in one, with no central server or database. A blockchain is an end-to-end network of computers known as nodes that share all of the network's data and code. So, if a device is connected to the blockchain, then that device will become a node in the network and can communicate with all the other computer nodes in the network. On the blockchain, the user device now has a copy of all the data and code. There are no longer any centralised servers. Simply a collection of computers that communicate with one another over the same network. The proposed methodology took online votes using a smart contract and calculated the time taken by the user to cast a vote. Proposed technology also calculates the load capacity when the number of users increases and the time taken to cast the vote.

The key factor of the proposed system is that it is implemented using a smart contract. The Ethereum blockchain allows to run code on the blockchain using the ethereum virtual machine (EVM) and a smart contract. The application's business logic is stored in smart contracts. This is where the decentralised component of the code will be written. Data can be written on blockchain by use of smart contracts, as well it also has the capability of conducting business logic. Solidity is a programming language whose syntax looks a lot like JavaScript, it is used to create smart contracts. Smart contracts on the blockchain work similarly to microservices on the internet. To visualise a smart contract, blockchain can be viewed as having two layers, one is a public ledger that stores data and can be seen as a database layer, other is a smart contract that helps to implement the business logic on the stored blockchain data.

Solidity is used to write the core contract code, which is how it works. The proposed work utilised the ganache encryption method. Truffle will provide an environment in which the EVM can run, whereas metamask will give an easy and safe mechanism to connect to blockchain-based applications. Figure 1 shows the proposed system architecture of the blockchain and smart contract enabled electronic voting system.

Figure 1 Decentralised application (DAPP) architecture (see online version for colours)



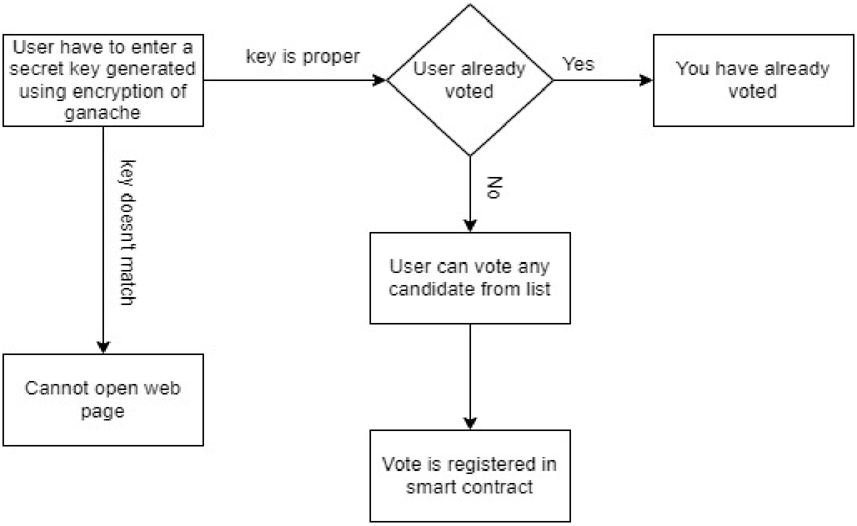
In the proposed method backend consists of blockchain as data will be stored in smart contracts which are written in solidity and solidity is built on top of the Ethereum blockchain network. To deploy any kind of smart contract, some amount of gas is required, similarly, whenever any functions written in the smart contract are invoked it costs i.e., some amount of gas. So, to spend gas, ganache is used, as ganache provides fake ether which can be used to deploy a smart contract on ganache local blockchain network.

The frontend part of the proposed application is built using HTML, CSS, JAVASCRIPT. HTML is used to structure the page, CSS is used to design the page, and in JavaScript React framework is used to provide the functionality for the user to vote.

If any user wants to cast a vote, it has to be connected to the local blockchain network. Frontend and backend are not connected yet, metamask is a chrome extension that provides functionality to connect the localhost to the blockchain network. Metamask requires a hashed string provided by ganache to connect to the active blockchain network. It connects all the pieces and now users can cast the vote by logging into metamask with their blockchain account and voting for their favorite candidate. Ganache also provides blockchain accounts that consist of an account address and an encrypted key which acts as a password to log into that account by metamask.

During elections, voters must log in using a safe and unique key obtained by the ganache encryption process. Users will be unable to access the web application if the key is wrong. If a user has previously voted, the user will be unable to vote again; otherwise, the user can view a list of candidates from which the user can vote, as illustrated in Figure 2. The vote will be recorded and saved in the smart contract when it is cast.

Figure 2 Voting procedure


User have to enter a secret key generated using encryption of ganache
key is proper
User already voted
Yes
You have already voted
key doesn't match
No
User can vote any candidate from list
Cannot open web page
Vote is registered in smart contract

Algorithms 1**: **E-Voting: Creating a data structure for the candidate

Input: ID of the candidate, Name of candidate, Initial vote count of candidate (initialized with 0)

Output: An object type of data structure for a candidate is created with a name, ID, and vote count attributes.

Step 1: Candidate ID, Name, and the vote count is provided to the smart contract and it creates a candidate structure for that ID, name, vote count.

```
    Struct Candidate
    {
        uint ID             //ID of candidate
        string name         //Name of candidate
        uint voteCount      //vote count of candidate
    }
```

Step 2: Multiple candidates can be created for the voting process in step 3.

Step 3: A constructor is called with the name of candidates which calls a function called addCandidate which takes string name as a parameter, it initiates candidate ID with 0 for the first candidate and increases the ID by 1 as more candidates are added. It also initializes the vote count as 0.

```
//initialize the constructor
    constructor ()
    {
        addCandidate("Donald trumph");      //Put the Name of Candidate
        addCandidate("Barack obama");
    }
    function addCandidate (string memory name) private
    {
        candidatesCount ++;                 //Increase the Count of Candidate
        candidates[candidatesCount] = Candidate (candidatesCount, name, 0); //Add another Candidate and
          Put Default Value
     }
```

4 Results and discussions

Results show the implementation of an e-voting system using the smart contract in blockchain technology. The performance of the system is evaluated using the hidden and visible voting gas cost for each user, the amount of time taken by the system to register a vote or reject a vote, and the accuracy of the system.

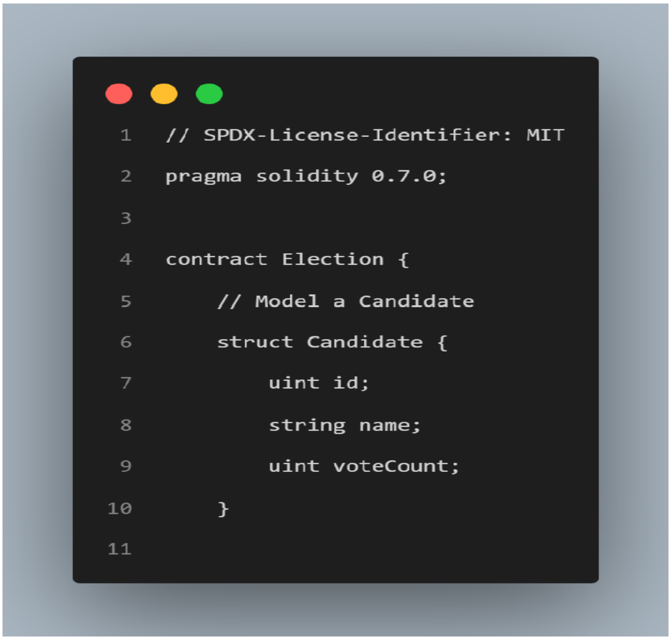
4.1 Implementation of smart contract

Every candidate that a user can vote for has 3 main attributes candidate ID, candidate name, and candidate vote count. Candidate ID is used to give unique identity attributes to each candidate which is shown in Figure 3. Candidate name is used to display on voting page and candidate vote count stores the number of votes for that particular candidate.

Algorithm 2: Voting function

```
If (user has voted already)
  {
     Reject the smart contract transaction
  }
Else
 {
     If (check candidate user voting for is valid)
       {
          register vote and increase vote count for that candidate
       }
     Else
       {
          reject the transaction
       }
  }
```

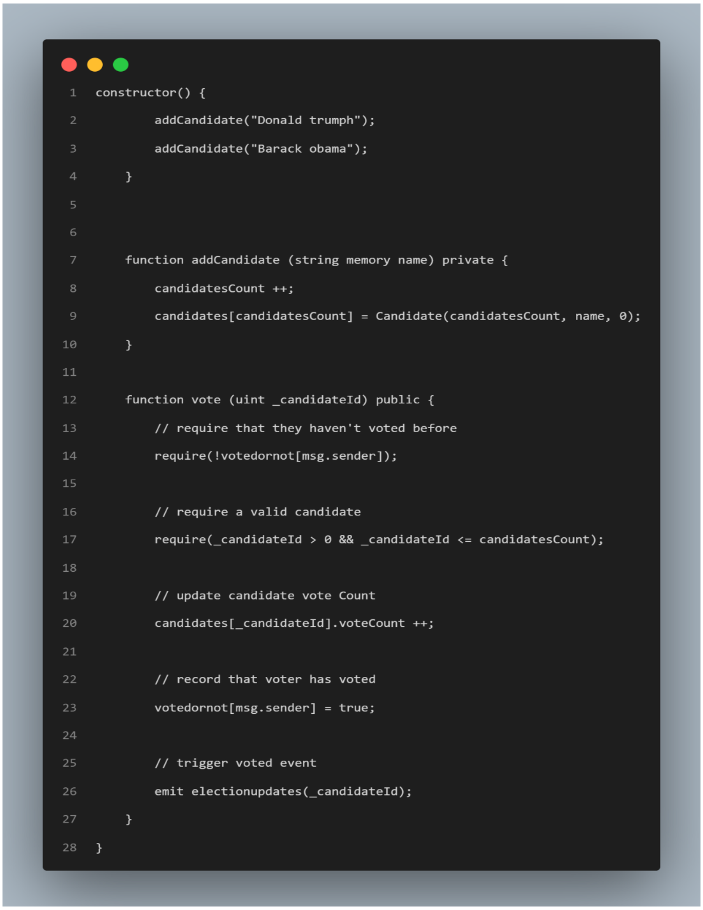
Figure 3 Data structure of a candidate in solidity (see online version for colours)



```
// SPDX-License-Identifier: MIT
pragma solidity 0.7.0;

contract Election {
    // Model a Candidate
    struct Candidate {
        uint id;
        string name;
        uint voteCount;
    }

```

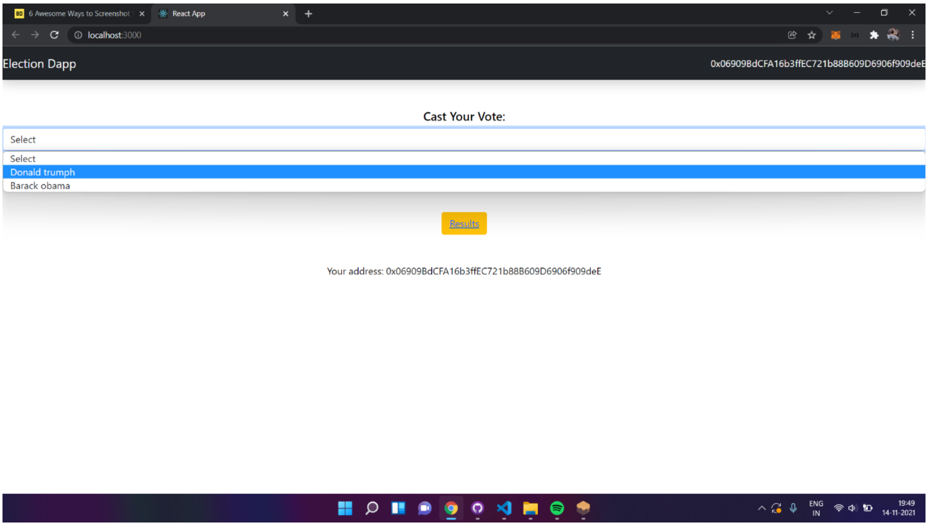
Figure 4 shows the construction of a database for the candidates. The candidate's name, candidate id, and vote count are initialised. The user can vote and the vote is checked to be valid only if the user had not voted before. In the case of a valid vote, the vote count of the candidate is incremented by 1 for which the user has voted.

Figure 4 Smart contract for voting (see online version for colours)



```
constructor() {
        addCandidate("Donald trumph");
        addCandidate("Barack obama");
    }


    function addCandidate (string memory name) private {
        candidatesCount ++;
        candidates[candidatesCount] = Candidate(candidatesCount, name, 0);
    }

    function vote (uint _candidateId) public {
        // require that they haven't voted before
        require(!votedornot[msg.sender]);

        // require a valid candidate
        require(_candidateId > 0 && _candidateId <= candidatesCount);

        // update candidate vote Count
        candidates[_candidateId].voteCount ++;

        // record that voter has voted
        votedornot[msg.sender] = true;

        // trigger voted event
        emit electionupdates(_candidateId);
    }
}
```

Figure 5 Voting interface (see online version for colours)



Election Dapp

0x06909BdCFA16b3ffEC721b88B609D6906f909deE

Cast Your Vote:

Select

Select
Donald trumph
Barack obama

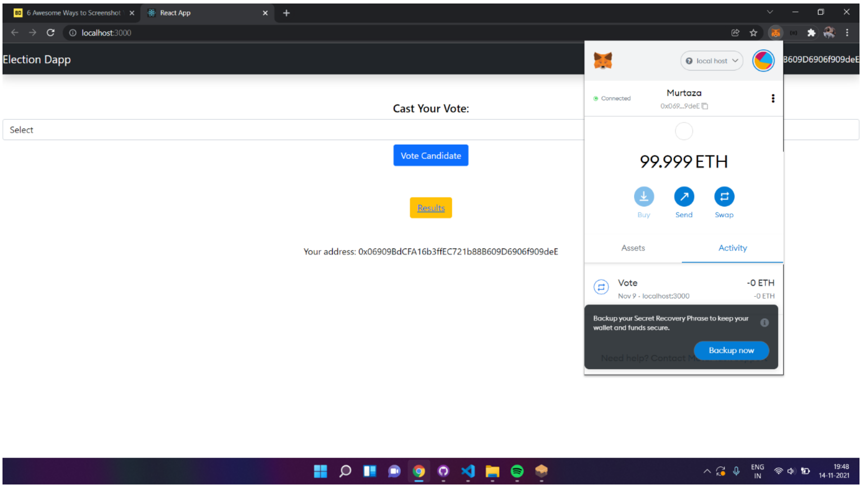
Results

Your address: 0x06909BdCFA16b3ffEC721b88B609D6906f909deE

Figure 5 shows the interface for the user to vote, the user needs to be connected to their blockchain account to perform the voting operation. They can connect to blockchain networks via the metamask chrome extension.

Figure 6 shows the account that is connected to the blockchain network that is hosted on localhost. Only after connecting to the blockchain network on which the application is hosted, the user can perform the voting operation.

Figure 6 Connecting to metamask (Connecting to local blockchain network) (see online version for colours)



Election Dapp

Cast Your Vote:

Select

Your address: 0x06909BdCFA16b3ffEC721b88B609D6906f909deE

Murtaza
0x069...9deE

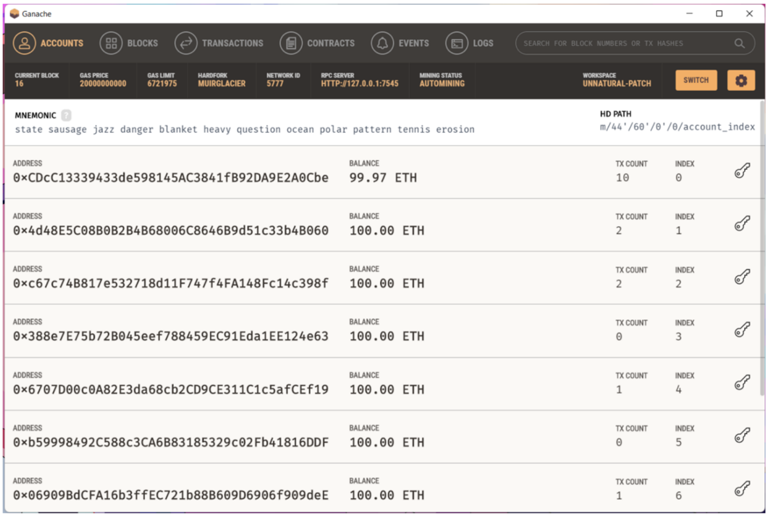
99.999 ETH

Vote -0 ETH
Nov 9 · localhost:3000 -0 ETH

Backup your Secret Recovery Phrase to keep your wallet and funds secure.

Ganache provides some blockchain account that contains fake ether which can be used as gas to deploy smart contract and perform the voting operation. In Figure 7, the account provided by ganache are visible with the amount of fake ether in them, it also shows several transactions done by that account address.

Figure 7 Ganache interface (see online version for colours)



CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
16	20000000000	6721975	MUIRGLACIER	5777	HTTP://127.0.0.1:7545	AUTOMINING	UNNATURAL-PATCH

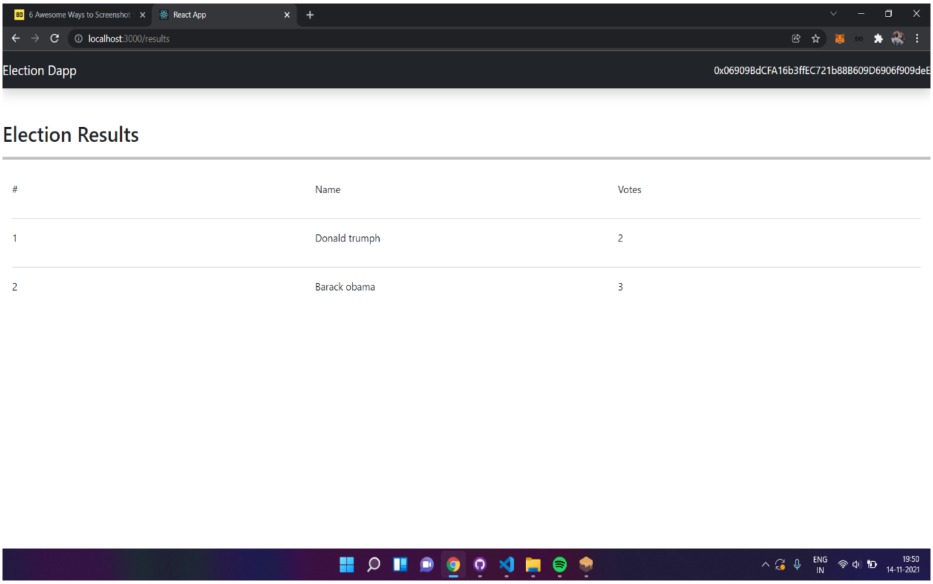
MNEMONIC
state sausage jazz danger blanket heavy question ocean polar pattern tennis erosion

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×CDcC13339433de598145AC3841fB92DA9E2A0Cbe	99.97 ETH	10	0
0×4d48E5C08B0B2B4B68006C8646B9d51c33b4B060	100.00 ETH	2	1
0×c67c74B817e532718d11F747f4FA148Fc14c398f	100.00 ETH	2	2
0×388e7E75b72B045eef788459EC91Eda1EE124e63	100.00 ETH	0	3
0×6707D00c0A82E3da68cb2CD9CE311C1c5afCEf19	100.00 ETH	1	4
0×b59998492C588c3CA6B83185329c02Fb41816DDF	100.00 ETH	0	5
0×06909BdCFA16b3fFEC721b88B609D6906f909deE	100.00 ETH	1	6

On clicking the result button visible in Figure 8, it redirects the application to the result page, where the user can see how many votes each candidate has received.

Figure 8 Result page (see online version for colours)



Election Dapp

0x06909BdCFA16b3ffEC721b88B609D6906f909deE

Election Results

#	Name	Votes
1	Donald trumph	2
2	Barack obama	3

4.2 Gas cost estimation

Table 1 shows data about deploying a smart contract on the blockchain network. The system has used only one account to deploy smart contracts that's why the Account address for contract deployment is the same for all 4 observations. It also shows the address of the smart contract that got created as well as the gas cost for deploying the contract.

Table 1 Gas cost for contract deployment on blockchain

Account address for contract deployment	*Gas cost used* (*Hidden*)	*Created contract address*
0xCDcC13339433de598145AC3841fB92DA9E2A0Cbe	186951	0xD83D0c84A191bc5465F71Ad67A8935d2e7842630
0xCDcC13339433de598145AC3841fB92DA9E2A0Cbe	385335	0xCdF157b7aD1d1df6bc919b9D972c77287Ee41658
0xCDcC13339433de598145AC3841fB92DA9E2A0Cbe	186951	0xc5c3Ac46CB33B877565b8f80C42Da2CA9A61FFE8
0xCDcC13339433de598145AC3841fB92DA9E2A0Cbe	385335	0x72814b0cBB2bA0Dc60570a626292349158E20640

Table 2 shows data about how much gas cost it takes to register a successful or unsuccessful vote. If the user has already voted then the vote gets rejected it also costs some gas amount.

Average gas cost for contract deployment (Hidden):

> Sum of Gas Cost for contract deployment/number of times contract was deployed = 1,144,572/4 = 286,143

Average gas cost for successful vote (Visible):

> Sum of Gas Cost for successful vote/number of successful votes = 337,464/6 = 56,244

Average gas cost for rejected vote (Hidden):

> Sum of Gas Cost for rejected votes/number of rejected votes = 89,415/4 = 22,354
>
> Percentage of Visible Cost for Deployment + Voting = Average cost for Successful votes/Sum of average of contract deployment + rejected vote + successful vote = (56244/364741) *100 = 15.42%
>
> Percentage of Hidden Cost for Deployment + Voting = Average cost for contract deployment + Average cost for rejected votes/Sum of average of contract deployment + rejected vote + successful vote = (308497/364741) *100 = 84.58%
>
> Percentage of Visible Cost Voting = Average of successful vote gas cost/Average of successful vote gas cost + Average of unsuccessful vote gas cost= (56244/78598) *100 = 71.55%
>
> Percentage of Hidden Cost Voting = Average of unsuccessful vote gas cost/Average of successful vote gas cost + Average of unsuccessful vote gas cost = (22354/78598) *100 = 28.45%

Table 2 Gas cost for voting

Account address	*Gas cost used for successful vote (Visible)*	*Gas cost for rejection of vote (Hidden)*
0x388e7E75b72B045eef788459EC91Eda1EE124e63	51244	–
0xb59998492C588c3CA6B83185329c02Fb41816DDF	–	22380
0x06909BdCFA16b3ffEC721b88B609D6906f909deE	51244	22349
0x6707D00c0A82E3da68cb2CD9CE311C1c5afCEf19	51244	–
0xc67c74B817e532718d11F747f4FA148Fc14c398f	51244	–
0xCDcC13339433de598145AC3841fB92DA9E2A0Cbe	66244	22349
0x4d48E5C08B0B2B4B68006C8646B9d51c33b4B060	66244	22337

4.3 Time efficiency

Table 3 contains the time taken by the system per vote concerning the address and the average execution time for the system to accept or reject the vote is 41/7 = 6 ms (approx.). Time efficiency tables show data about how much time it took the system to execute the voting function. The time mentioned in the table is in milliseconds. It also shows the address of the account used to vote. This result is obtained on machine Intel(R) Core (TM) i5-10300H CPU@ 2.50GHz Processor with 8 gigabytes of RAM.

Table 3 System time per vote

Address used to vote	*Time taken for vote to get accepted or rejected by the system* (in Milli Seconds)
0x388e7E75b72B045eef788459EC91Eda1EE124e63	4
0xb59998492C588c3CA6B83185329c02Fb41816DDF	10
0x06909BdCFA16b3ffEC721b88B609D6906f909deE	12
0x6707D00c0A82E3da68cb2CD9CE311C1c5afCEf19	3
0xCDcC13339433de598145AC3841fB92DA9E2A0Cbe	7
0x4d48E5C08B0B2B4B68006C8646B9d51c33b4B060	7
0xc67c74B817e532718d11F747f4FA148Fc14c398f	8

5 Conclusion

A secure and cost-efficient E-Voting is a mandatory requirement of every democratic nation and it should be solved by implementing the E-Voting System through some reliable and trustworthy technologies. In this contribution, a blockchain and the smart contract-based solution is proposed to make voting easier, secure, efficient, and inexpensive. This work used ganache, metamask, and truffle framework as the blockchain-based technology to achieve the smooth functioning of secure voting. The application is implemented using HTML, CSS, JavaScript by integrating with blockchain technology using Web 3.0 Application Programming Interface (API) for the frontend. The smart contract is written in solidity which is based on Ethereum and the contract is deployed on a blockchain network using ganache. The algorithm for E-voting smart contract is implemented and the voter can interact with the interface using browser by using their existing account and connecting it via metamask to the blockchain network and voting for the candidate of their choice. The algorithm used for encryption is keccak256 which is a cryptographic function built into solidity. The performance of the blockchain-based E-Voting system is analyzed using execution time and gas cost for voting and contract deployment. The execution time calculated for registration of successful or rejected votes is 6ms per vote. The visible and hidden gas cost for voting is 15.42% and 84.58% respectively of total gas cost, hidden gas cost consists of rejected votes and contract deployment while visible gas cost consists of successfully registered votes.

6 Future scope

To implement E-Voting System based on blockchain on large scale for a country like INDIA, voters can be provided with their blockchain accounts along with their Aadhar card once they are eligible for voting. A unique cryptocurrency can also be made for the voting process, that cryptocurrency will only be used for voting. The challenge can be that every voter should have a smart device like mobile or desktop with an active internet connection to cast a vote. The work may be extended to reduce the gas cost for bulk voting.

Funding

This study did not receive any funding in any form.

Conflicts of interest/Competing interests

The authors declare that there is no conflict of interest.

Availability of data and material

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Code availability (software application or custom code)

Code for blockchain implementations are available on request due to privacy or other restrictions.

Authors contributions

Murtaza Bagwala, Alakmar Barwaniwala, and Manas Garg contributed to the design and implementation of the research. Kailash Chandra Bandhu and Ratnesh Litoriya performed the analysis of the results and contributed to the writing of the manuscript. All authors discussed the results and contributed in the final manuscript preparation.

References

Agbesi, S. (2020) 'Examining voters' intention to use internet voting system: a case of Ghana', *International Journal of Electronic Governance*, Vol. 12, No. 1, p.57, doi: 10.1504/IJEG.2020.106997.

Agrawal, T.K., Angelis, J., Khilji, W.A., Kalaiarasan, R. and Wiktorsson, M. (2022) 'Demonstration of a blockchain-based framework using smart contracts for supply chain collaboration', *International Journal of Production Research*, pp.1–20, doi: 10.1080/00207543.2022.2039413.

Alharby, M., Aldweesh, A. and Van Moorsel, A. (2018) 'Blockchain-based smart contracts: a systematic mapping study of academic research 2018', *2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB)*, IEEE, Fuzhou, China, pp.1–6, doi: 10.1109/Iccbb.2018.8756390.

Al-madani, A.M., Gaikwad, A.T., Mahale, V. and Ahmed, Z.A.T. (2020) 'Decentralized E-voting system based on smart contract by using blockchain technology', *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing* (*ICSIDEMPC*), IEEE, pp.176–180, doi: 10.1109/ICSIDEMPC49020.2020.9299581.

Bodkhe, U., Bhattacharya, P., Tanwar, S., Tyagi, S., Kumar, N. and Obaidat, M.S. (2019) 'BloHosT: blockchain enabled smart tourism and hospitality management', *2019 International Conference on Computer, Information and Telecommunication Systems* (*CITS*), Beijing, China, pp.237–241, doi: 10.1109/CITS.2019.8862001.

Buterin, V. (2014) *Ethereum White Paper*, Etherum White Paper, January, pp.1–36.

Cai, W., Wang, Z., Ernst, J.B., Hong, Z., Feng, C. and Leung, V.C.M. (2018) 'Decentralized applications: the blockchain-empowered software system', *IEEE Access*, Vol. 6, pp.53019–53033, doi: 10.1109/ACCESS.2018.2870644.

Dubey, R., Gunasekaran, A., Bryde, D.J., Dwivedi, Y.K. and Papadopoulos, T0 (2020) 'Blockchain technology for enhancing swift-trust, collaboration and resilience within a humanitarian supply chain setting', *International Journal of Production Research*, Vol. 58, No. 11, pp.3381–3398, doi: 10.1080/00207543.2020.1722860.

Gao, S., Zheng, D., Guo, R., Jing, C. and Hu, C. (2019) 'An anti-quantum E-voting protocol in blockchain with audit function', *IEEE Access*, Vol. 7, pp.115304–115316, doi: 10.1109/ ACCESS.2019.2935895. Grech, L. and Camilleri, A.F. (2017) *Blockchain in Education*, doi: 10.2760/60649.

Guo, Y. and Liang, C. (2016) 'Blockchain application and outlook in the banking industry', *Financial Innovation*, Vol. 2, No. 1, p.24, doi: 10.1186/s40854-016-0034-9.

Gupta, R., Tanwar, S., Al-Turjman, F., Italiya, P., Nauman, A. and Kim, S.W. (2020) 'Smart contract privacy protection using AI in cyber-physical systems: tools, techniques and challenges', *IEEE Access*, Vol. 8, pp.24746–24772, doi: 10.1109/ACCESS.2020.2970576.

Gupta, R., Shukla, A. and Tanwar, S. (2020) 'AaYusH: a smart contract-based telesurgery system for healthcare 4.0', *2020 IEEE International Conference on Communications Workshops, ICC Workshops 2020 – Proceedings*, Dublin, Ireland, doi: 10.1109/ ICCWorkshops49005.2020.9145044.

Hjalmarsson, F.P., Hreiðarsson, G.K., Hamdaqa, M. and Hjálmtýsson, G. (2018) 'Blockchain-based E-voting system', *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, IEEE, San Francisco, CA, USA, pp.983–986, doi: 10.1109/Cloud.2018.00151.

Jafar, U., Aziz, M.J.A. and Shukur, Z. (2021) 'Blockchain for electronic voting system–review and open research challenges', *Sensors*, Vol. 21, No. 17, p.5874, doi: 10.3390/s21175874.

Jaiswal, S., Dalvi, Y. and Sharma, P. (2021) 'E-voting using blockchain', *International Journal of Engineering Research and Technology*, Vol. 10, No. 3, pp.278–280.

Kabra, N., Bhattacharya, P., Tanwar, S. and Tyagi, S. (2020) 'MudraChain: blockchain-based framework for automated cheque clearance in financial institutions', *Future Generation Computer Systems*, Vol. 102, pp.574–587, doi: doi.org/10.1016/j.future.2019.08.035.

Khan, K.M., Arshad, J. and Khan, M.M. (2020) 'Investigating performance constraints for blockchain based secure e-voting system', *Future Generation Computer Systems*, Vol. 105, pp.13–26, doi: 10.1016/j. future.2019.11.005.

Khan, S.N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E. and Bani-Hani, A. (2021) 'Blockchain smart contracts: applications, challenges, and future trends', *Peer-to-Peer Networking and Applications*, Vol. 14, No. 5, pp.2901–2925, doi: 10.1007/s12083-021-01127-0.

Lai, W-J., Hsieh, Y-c., Hsueh, C-W. and Wu, J-L. (2018) 'DATE: a decentralized, anonymous, and transparent E-voting system', in *2018 1st IEEE International Conference on Hot Information-Centric Networking* (*HotICN*), IEEE, Shenzhen, China, pp.24–29, doi: 10.1109/ HOTICN.2018.8605994.

Leeming, G., Cunningham, J. and Ainsworth, J. (2019) 'A ledger of me: personalizing healthcare using blockchain technology', *Frontiers in Medicine*, p.6, doi: 10.3389/fmed.2019.00171.

Mengelkamp, E., Notheisen, B., Beer, C., Dauer, D. and Weinhardt, C. (2018) 'A blockchain-based smart grid: towards sustainable local energy markets', *Computer Science – Research and Development*, Vol. 33, Nos. 1–2, pp.207–214, doi: 10.1007/s00450-017-0360-9.

Pandey, P. and Litoriya, R. (2020a) 'Implementing healthcare services on a large scale: challenges and remedies based on blockchain technology', *Health Policy and Technology*, Vol. 9, No. 1, pp.69–78, doi: 10.1016/j. hlpt.2020.01.004.

Pandey, P. and Litoriya, R. (2020b) 'Promoting trustless computation through blockchain technology', *National Academy Science Letters*, Vol. 44, pp.225–231, doi: 10.1007/s40009-020-00978-0.

Pandey, P. and Litoriya, R. (2020c) 'Securing and authenticating healthcare records through blockchain technology', *Cryptologia*, Taylor, and Francis, Vol. 44, No. 4, pp.341–356, doi: 10.1080/01611194.2019.1706060.

Pandey, P. and Litoriya, R. (2020d) 'Securing E-health networks from counterfeit medicine penetration using blockchain', *Wireless Personal Communications*, Vol. 117, pp.7–25, doi: 10.1007/s11277-020-07041-7.

Pandey, P. and Litoriya, R. (2021) 'Technology intervention for preventing COVID-19 outbreak', *Information Technology and People*, doi: 10.1108/ITP-05-2020-0298.

Qashlan, A., Nanda, P., He, X. and Mohanty, M. (2021) 'Privacy-preserving mechanism in smart home using blockchain', *IEEE Access*, Vol. 9, pp.103651–103669, doi: 10.1109/ACCESS.2021.3098795.

Shahzad, B. and Crowcroft, J. (2019) 'Trustworthy electronic voting using adjusted blockchain technology', *IEEE Access*, Vol. 7, pp.24477–24488, doi: 10.1109/ACCESS.2019.2895670.

Shejwal, P., Jadhav, M.S., Gaikwad, A.B., Nanaware, N.N. and Shikalgar, N.S. (2019) 'E-voting using blockchain technology', *International Journal of Scientific Development and Research*, Vol. 4, No. 5, pp.583–588.

Soner, S., Litoriya, R. and Pandey, P. (2021) 'Exploring blockchain and smart contract technology for reliable and secure land registration and record management', *Wireless Personal Communications*, Vol. 121, No. 1, pp.2495–2509, doi: 10.1007/s11277-021-08833-1.

Soner, S., Litoriya, R. and Pandey, P. (2022) 'Integrating blockchain technology with ioT and ML to avoid road accidents caused by drunk driving', *Wireless Personal Communications*, doi: 10.1007/s11277-022-09695-x.

Verma, A., Nawaz, S., Singh, S.K. and Pandey, P. (2021) 'Importance of 5G-enabled IoT for industrial automation', *Blockchain for 5G-Enabled IoT-The New Wave for Industrial Automation*, Springer, Ahmedabad, India.

Wijaya, D.A., Liu, J.K., Suwarsono, D.A. and Zhang, P. (2017) 'A new blockchain-based value-added tax system', in Okamoto T, *et al.* (Eds.): *Provable Security*, Springer (Lecture Notes in Computer Science), Xi'an, China, pp.471–486, doi: 10.1007/978-3-319-68637-0_28.

Yi, H. (2019) 'Securing e-voting based on blockchain in P2P network', *EURASIP Journal on Wireless Communications and Networking*, Vol. 2019, No. 1, p.137, doi: 10.1186/s13638-019-1473-6.