ONLINE VOTING APPLICATION USING ETHEREUM BLOCKCHAIN

Shalini Shukla
Department of Information Science and Engineering
PES Institute of Technology
Bangalore, India

Shashank D O
Department of Information Science and Engineering
PES Institute of Technology
Bangalore, India

Thasmiya A N
Department of Information Science and Engineering
PES Institute of Technology
Bangalore, India

Dr. Mamatha H R
Department of Computer Science and Engineering
PES University
Bangalore, India

Abstract— **Voting is an important part of the administration of a country. Votes are still being carried out by physically going to voting booths. This process doesn't guarantee security and cases of tampering has been observed. This paper aims at removing these issues in the voting process by making it online and using the technology, Blockchain. Blockchain uses encryption and hashing to make every vote secure. In this case, one vote is considered as a transaction. A peer to peer network is created to create a private blockchain that share this distributed ledger having voting transaction. The application is designed in such a way so that the intricacies of the underlying architecture is hidden from the user. Each voter is uniquely identified by Government approved Aadhar number. The application makes use of this number to make sure that each voter gets only one chance to vote. When the vote gets submitted as a transaction then all the peers get synch up. Since each peer is associated with a public and private key the votes are encrypted and hashed and added to the blockchain to increase security and form a chain of blocks. Votes cannot be tracked back to the voter. In this paper, a peer to peer network is created having minimum three peers. Since voting is made online, it is expected that this paper will increase the voter turnouts. The scalability of the blockchain application depends on the secondary memory limit of the peer.**

Index Terms— **Blockchain, Ethereum, Geth, IPC, Proof of Work(PoW), p2p, RPC, smart contract, solidity, truffle**

I. Introduction

Individuals utilize the term 'blockchain innovation' to mean diverse things, and it can be befuddling. Here and there they are discussing The Bitcoin Blockchain, now and again it's other virtual monetary standards, in some cases it's smart contracts. However, they are discussing dispersed records, i.e. a rundown of exchanges that is shared among various PCs, as opposed to being put away on a centralized server. Blockchain technology has been present since the 80's, the reason why this technology is more talked about these days is due to bitcoin. Bitcoin was developed in 2009 [1], ever since then, bitcoin has been considered as the first example of the digital asset (digital currency) to be used for making financial transactions, even though it has no essential value and no centralized controller for managing the transactions. Further, the importance should be noted that the underlying technology on which it was developed is blockchain Equations. Election is an important process in a democratic country. In this day and age, far reaching question towards the administration and impedance in nations' procedures by outside world have made the just procedure of voting more basic than any time in recent memory. Majority nations have been encountering tyrannical administrations which have presented far reaching dread among their people. Individuals had their human rights damaged and their basic flexibilities gave by their constitution taken away. In such a climate, having a reasonable and straightforward election system is something that is vital for the opportunity a great many people appreciate today.

978-1-5386-5314-2/18/$31.00 ©2018 IEEE

The traps of the present arrangement of poll voting are being exploited by individuals or associations hoping to pick up control. The problems associated with elections can be avoided if the vote counting process was transparent, verifiable and fair. The current system provides anonymity to the voted but it is not considered to be transparent. People are expected to trust the results that is announced by the government when it comes to elections. There are many frauds involved in voting like ballot stuffing, booth capturing and voter fraud. All this is making the process of voting really hard. Another problem involved when considering Indian elections is that the voting centers are far, and the voter has to physically go to the voting centers in order to vote so the ratio of the amount of people who cast their vote and the people who are eligible to vote and is reducing drastically. In such situations, it is desirable to have a system to overcome all the above problems.

A. Blockchain Technology

A blockchain is a public ledger which is digitized and decentralized for all cryptocurrency transactions. It can be considered as a linked list of transactions in the form of blocks [2]. Each block is connected to the previous block using the hash of the previous block. This is how a blockchain is formed. The blockchain is secured cryptographically. It makes use of peer to peer networking. This blockchain is a distributed shared ledger between the peers in the network. Each peer is associated with a public and private key. The public key of a peer is known to all the other peers in the network. When a transaction is submitted it is not immediately added to the blockchain, it is considered as a pending transaction. All the peers in the network have some pending transactions. These pending transactions form a block that has not yet been added to the blockchain. Now the question is which block will be added next to the blockchain? [3]. All the peers in the network have to solve a puzzle. Given a set of pending transactions they need to generate a random number (nonce) such that the final hash begins with seventeen zeroes. This is a compute intensive process and is called mining. The node that solves this puzzle first advertises the transaction, nonce to the whole network and all the other nodes verify this information and adds to the blockchain. The main concern here is the transactions happening between people is stored on a network of strangers, how do we trust the strangers with our transaction information. Blockchain aims to remove the entire concept of trust. Each transaction has a digital signature associated with it and this is different for every transaction. A signature is a function of the message and the private key of the peer. A digital signature is a way of proving that you have the password without revealing the password. Other peers verify this transaction using the transaction message and the public key of the peer. There are many implementations of blockchain technology for example hyperledger by IBM, ethereum etc. We shall be talking more about ethereum as this technology has been used in this paper. Ethereum, an open-source, distributed, is a blockchain-based computing platform for developing blockchain-based applications, also called as DApps (Decentralized Applications) [4]. A smart contract is a logical code that facilitates the exchange of shares, property, money etc. When blockchain is running, a smart contract will run automatically and executes when few specific conditions and cases are met. Once a smart contract is deployed it can't be changed. Hence a smart contract has data and code and since it runs on blockchain it is resistant to censorship, fraud etc.

B. Ethereum Blockchain

The goal of Ethereum is an alternative application to build a decentralized application which will be an open source [5]. Ethereum does this by building an abstract foundation layer for developing blockchain applications. With the development of Ethereum platform, it enables anybody to express "smart contracts", and decentralized applications where they can make their own agreement standards and rationale for ownership, exchange configurations and state transition functionalities [6]. In Ethereum, the state is comprised of items called as "accounts", a record is a 20-byte address, and the state transition functionalities are utilized to exchange esteems and data between the records. Ethereum accounts are comprised of 4 fields, a nonce, which is counter use to keep track of unique transactions, account's "ether" balance, "contract code" and accounts "storage". Ethereum nodes store and process data using Ethereum Virtual machine (EVM) [7]. A smart contract is written in Solidity language. This is compiled into EVM bytecode and run onto the node. Smart contract is deployed on blockchain network using Truffle and geth – go implementation of Ethereum, is used to bring up our private blockchain network.

This paper is structured as follows. In section 2, the paper discusses about the background work related to blockchain and voting. Section 3 discusses about the methodology that is used to execute this idea. Section 4 deals with the results and discussion. In Section 5, the paper talks about drawbacks and limitations of the present architecture. Section 6 tackles on future enhancements.

II. MOTIVATION AND RELATED WORKS

The motivation behind this paper is to use the concept of blockchain to provide secure e-voting. The meaning of true democracy is to make sure that all its citizens have equal rights to choose their representative candidate in an unbiased fashion. Acts like vote tampering, booth capturing go against these ideas which leads to decisions be corrupted and controlled. Inorder to make the voting process secured and protected, we use the concepts of encryption and hashing. Physically going to the voting booths to vote is not very feasible if the citizen is someplace else and wants to vote for the elections. This leads to increase in the count of the voter turnups. To understand the core concepts and technology behind the development of blockchain, the Ethereum White Paper [8] was used as an initial reference. It constitutes concepts such as the invention of bitcoin, bitcoin is also the state transition system, the concept of mining, alternative use of blockchain, smart contracts for scripting, and then it talks about the Ethereum technology for the development of blockchain-based applications, Ethereum accounts, messages and transactions.

E-voting right now broadly utilized by a few nations in the world, for instance in Estonia [9]. The nation has been utilizing the e-voting framework since 2005 and in 2007 directed on the online voting and was the first nation on the planet to lead on the online voting. From that point forward, a lawfully restricting online voting framework has been executed in different associations and nations, for example, the Austrian Federation of Students, Switzerland [10], the Netherlands, Norway, etc. Yet, despite everything, it has security issues and the selection is often cancelled. In spite of the fact that getting a great deal of consideration, online voting framework is as yet not generally done in different nations around the world. The traditional voting framework has a few issues experienced when overseen by an association that has full control over the framework and database, along these lines the association can mess with the database, and when the database changes the follows can be effortlessly eliminated.

The solution is to make the database open, the database claimed by numerous clients, which is helpful to look at if there are any inconsistencies. The answer for the e-voting framework is perfect with utilizing blockchain innovation. Blockchain innovation permits in help of e-voting applications. Each voter's vote fills in as a transaction that can be made into blockchain. In this way, everybody can favor the last estimation in light of the open blockchain review trail, the vote check can be confirmed that no information is changed or erased nor is there any unapproved information entered in the blockchain.

The concept of e-voting is not new, there are many applications that have centralized servers and storage models. Countries like Estonia uses an online voting system while contesting elections [11]. The concept of e-voting was started in the year 2001 and started by the national authorities. This system is in use in the current times with various modifications in their code base. It is considered to be robust, transparent and reliable. The application is in the form of both a website as well as a desktop application. Since the infrastructure that they have used is centralized this is the major drawback that can be seen. Problems like single point of failure, hijacking takes place. This is also prone to attacks like DDOS (Distributed Denial of Service) wherein large amount of traffic is sent to the server thereby clogging the pathway for the right data. So hence the service provided by the server goes down.

Taking a look at the work that has been done in blockchain based voting application, there is a start-up called followmyvote [12]. In this application, voters are made anonymous and then they count the number of votes and they apply their mathematical formula that is used to distinguish between the valid and invalid voters. This process is really sophisticated and tends to create delay. This start-up has a long way to go to into production. Our application solves this problem by using the national identity of the voter to find out if he is a valid voter or not. There is also a one-time password that is sent out to his contact details to verify him.

In recent times, the government of Moscow is currently testing the blockchain infrastructure for their voting system [13]. In the words of the government - "The introduction of this technology will make the voting in the [Active Citizen initiative] even more open: it will be difficult to say that the administration incorrectly interprets the answers, changes the results of the voting when the citizens themselves can verify this information." This has attracted nations like Russia as well. This implementation is right now in the testing phase.

In [14], the authors have deployed their voting application onto the Rinkeby Network [15] which gives away free test ethers inorder to deploy applications on it. The blockchain that they have used is also ethreum. Their smart contract takes care of Voter validation using the smart contract itself. Their smart contract takes care if the voter has the rights to vote as well as if the voter can vote for a candidate only once. The disadvantage of this network is that they are currently depending on third party networks like Rinkeby and using test ethers to power up their application which is unlike our application that is built on a private blockchain.

In [16], the implementation is in such a way that they are operating with two blockchain's – one maintains the vote information and the other has voter information. And there are various levels of voting, one is at the municipality level, other is at the center level. Such kind of architecture brings about extra complexity in the network and is not often reliable and far from live production.

III. Proposed Methodology

As discussed, the blockchain technology that we have used here is Ethereum. The power of smart contracts provides huge range of use cases. Ethereum has a huge open source community having large varieties of software's that can be used together to build a secure distributed application. Considering the underlying security that Ethereum provides ranging from digital signatures to hashing, it is difficult to change the source code of the given software.

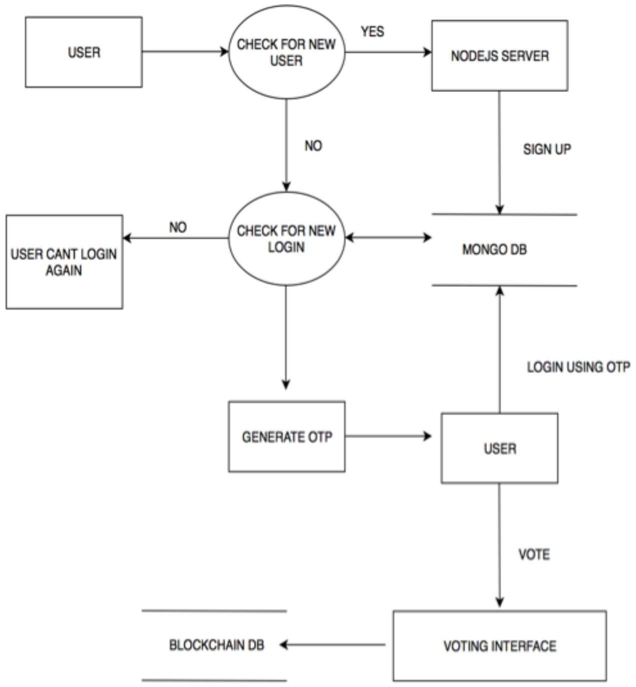


Fig.1. System Design

Ethers is the currency that is used to fuel ethereum network, in some terms it is also referred to as gas. All operations that you perform on this network requires some amount of gas to get it running. In our infrastructure, each vote is taken as one transaction, so whenever one vote is casted, a transaction is submitted to the blockchain network. To submit transactions the voter need, provide some amount of gas. In blockchain terminologies, you have a peer and a user which is the voter of our application. Peers are the miners that mine blocks or transactions that are submitted. A block is a set of transactions. These blocks are continuously mined by these peers. In our infrastructure, we have set up a private blockchain network so that we can decide as to what peers we want to connect to the network. Any peer that is routable and can be reached can be added to the peer. This is one distinction between a private and a public blockchain. All miners that are present in the network should have high computational power and memory because mining is a compute intensive process that involves all the miners to solve a puzzle inorder to win a reward in the form of ethers or bitcoin in bitcoin networks.

On our private blockchain network built using ethereum, we deployed our smart contract. This is the business logic that governs the network. It is irreversible, once deployed it can't be changed. Smart contracts are written in languages like Serpent, Solidity, Viper etc. We have used Solidity to write our smart contract. Solidity is a combination of languages like C++ and JavaScript. A smart contract deployment is done on the private blockchain and that is supported by the peers. As and when new peers are added to the network, they start executing the functions of the contract like the other peers.

Considering the problems associated with current election system of India such as vote tampering, standing in long lines to cast votes, booth capturing etc. We are focused on eradicating such problems and bring about transparency, authentication in our voting procedure. We need to make sure that whoever is voting is a valid candidate. We do this by verifying the Aadhar details of the individual and by using one time password (OTP). The voter is initially desired to sign up in the application. These details will be stored in MongoDB as backend. Inorder to cast his vote he needs to login using the details that he provided while signing up. A OTP will be generated and sent to him on his mobile number. Once verified he is taken to the voting portal where he can cast his vote. Once the user has logged in, he can't log in again as his data will be entered in the backend MongoDB. Once the user is directed to the voting portal, the application makes sure that once he is in the portal he can vote only once to only one candidate. So as described in the system design of this infrastructure, initially user has to sign up to the application and then log in, checks like login without sign up and OTP verification is taken care of by the application. All the contents of the blocks in the blockchain, all its necessary information is stored in a blockchain database that is implicit to ethereum and is the same for all the blocks that are in the blockchain as they all share the same distributed ledger. This is shown in the system architecture in the figure 1. As per system architecture, MongoDB and Blockchain are storage entities. The user logs on / logs in to the voting portal and the user detail is added into MongoDB if not present and OTP is generated and sent to the user, on correct submission of OTP, the user will be able to cast his vote to the candidate of his choice. The candidate details and the number of votes the candidate has received is stored in the blockchain.

Issues like vote tampering is taken care of by the blockchain infrastructure. Since the chain of blocks that are present with each of the peers is the same since we are dealing with the same distributed ledger, block (n+1) is connected to the block (n) which is the block before it is using the hash of the previous block this is hash(n).
Hash of a block is a function of:

Hash (block (n+1)) = f (timestamp+ hash(block(n)) + payload+ version of the software+ hash (Merkel root) + target+ Nonce)

So, if block(n) is tampered with, its hash changes, which means hash (block(n+1)) changes and then hash(block(n+2)) will also change and so on and so forth. This means that all the blocks after the modified block will have to change and each peer should calculate this individually which is compute intensive process so hence block tampering is not possible in such an infrastructure. Target is the difficulty value that is specified in the genesis file which specifies how difficult the puzzle is going to be for the peers to mine. Nonce is the random number that will be generated by the peers during mining. Whichever peer is able to generate the correct nonce will be rewarded. This is a part of the puzzle that peers solve. This is used for adding new blocks in the blockchain.

The block contents can be seen by using its hash as the identifier as a hash of one block is unique in a blockchain. The contents of one block consists of the following information:

1. Block hash: This is the hash generated using the information about the hash of the previous block, the timestamp, the version of the software used, the Merkel root or the binary hashing of all the transactions that are present inside the blocks of the blockchain, the difficulty value that is specified in the genesis file and the nonce or the random number that is generated in the mining process.
2. Block Number: This is the next consecutive number in the blockchain.
3. From: This refers to the public key of the voter who has performed the transaction which is the vote in this case. As discussed, every voter is associated with a set of public, private keys.
4. Gas Used: This refers to the amount of gas or ethers used to complete this transaction. This is supplied by the voter who has performed transaction.
5. Status: This field gives information about whether the transaction is successful or not, status 1 stands for a successful transaction and 0 if unsuccessful.
6. To: This is the public address of the smart contract to which the transaction is submitted to.
7. Transaction Hash: It is an identifier that uniquely identifies a particular transaction.

Coming over to our smart contract, there are four functions; function Voting is a constructor which is used to initialize the candidate list who are standing for the election. The function totalVotesFor() is like a getter function that is used to get the total number of votes that a candidate has received. The function voteForCandidate() is used to vote for the candidate that you desire. It gets the initial value of the vote for that candidate and then increments it. The function validCandidate() is used to check for whether the candidate that you vote for is in the candidate list.

The pseudo code for our smart contract is as follows:

```
contract Voting:

  hashMap votesReceived[candidateName => votesReceived]
 candidateList[]


# getter function to return the number of votes a candidate
#recieves
function totalVotesFor(candidateName):
Input: Candidate Name
Output: Votes Received
        return votesReceived[candidateName]


# increments vote for a candidate
function voteForCandidate(candidateName):
Input: Candidate Name
        votesReceived[candidateName] ++


#checks if the candidate is valid
function validCandidate(candidateName):
Input: Candidate Name
Output: Boolean Value
if candidateName is in candidateList:
        return True
else
        return False
```

For deploying the smart contract on our private blockchain we have used Ethereum Truffles which is a Development IDE that has capability of debugging, testing, creating smart contracts, deploying it on any desired network aiming to make things easier for a blockchain developer.

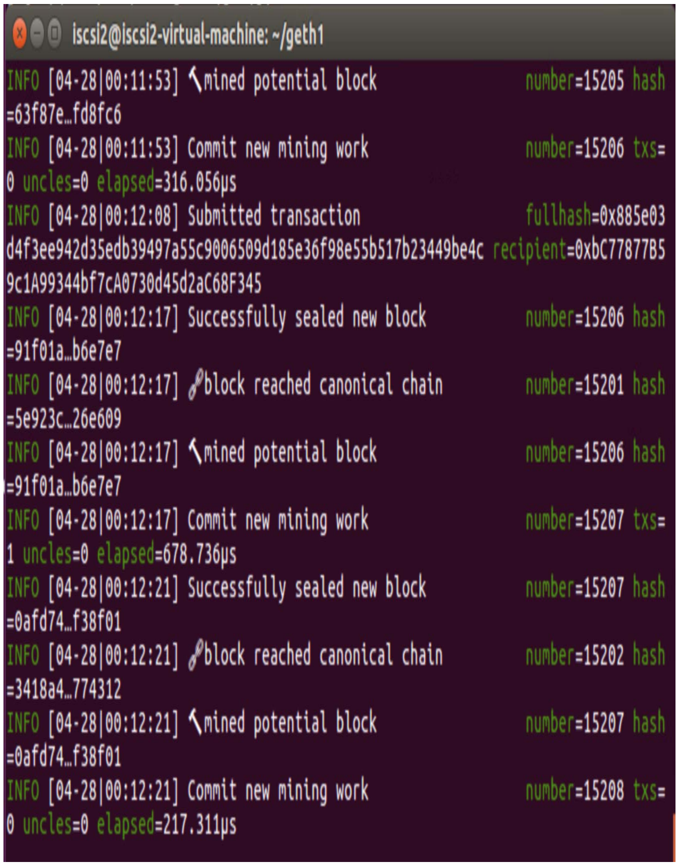
The peers have to mine continuously. The backend has to be up and running at all times. Peers can be continuously added and removed by the network. If a peer is compromised, damage can’t be done, because even if the data is seen the vote can’t be traced back to the user who has voted. When a transaction is submitted it a hash of the block is generated immediately by the mining peers and added to the blockchain. As discussed, the ability to mine depends on the difficulty value in your genesis file that is used to initialize the first block in your blockchain.

The genesis file consists of the following parameter:

1. ChainID: It refers to your chain identifier that is used for protection against replay. The other parameters are needed for chain versioning and forking.
2. Difficulty: This is an important part of the file. It is indication of how tough the mining process is made. More the value more difficult it is to mine. For our private blockchain we have used minimum value so

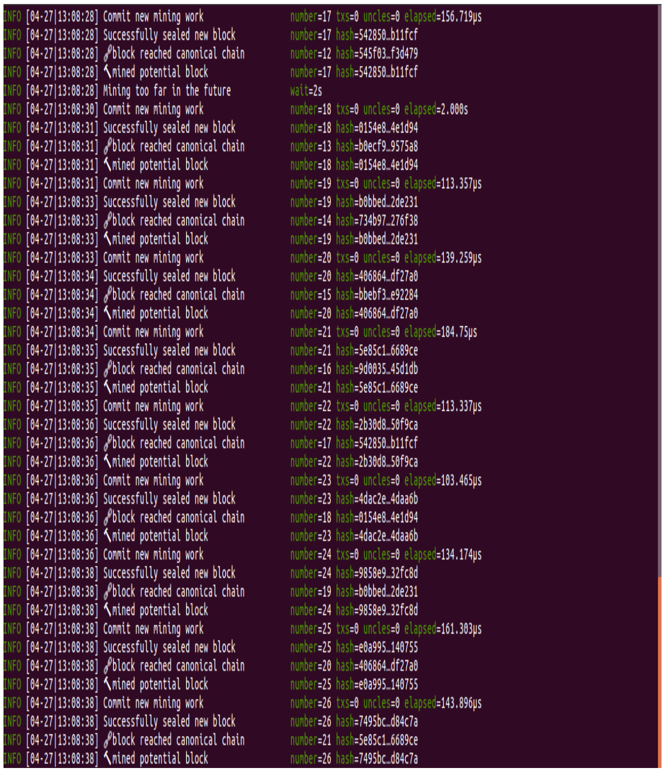
that we can easily mine the blocks faster and quickly thereby generating more ethers for the working of the application.

3. gasLimit: This refers to the maximum amount of ethers a block can use during mining. This value is recommended to be set high. Since the difficulty value is kept low, the gas limit should be kept high, so the blocks can be mined efficiently without slowing the application network.
4. Alloc: This field is not mandatory is used to assign some minimum balance for the geth accounts that you create.



```
INFO [04-28|00:11:53] 🔨mined potential block              number=15205 hash
=63f87e…fd8fc6
INFO [04-28|00:11:53] Commit new mining work               number=15206 txs=
0 uncles=0 elapsed=316.056µs
INFO [04-28|00:12:08] Submitted transaction                fullhash=0x885e03
d4f3ee942d35edb39497a55c9006509d185e36f98e55b517b23449be4c recipient=0xbC77877B5
9c1A99344bf7cA0730d45d2aC68F345
INFO [04-28|00:12:17] Successfully sealed new block        number=15206 hash
=91f01a…b6e7e7
INFO [04-28|00:12:17] 🔗block reached canonical chain       number=15201 hash
=5e923c…26e609
INFO [04-28|00:12:17] 🔨mined potential block              number=15206 hash
=91f01a…b6e7e7
INFO [04-28|00:12:17] Commit new mining work               number=15207 txs=
1 uncles=0 elapsed=678.736µs
INFO [04-28|00:12:21] Successfully sealed new block        number=15207 hash
=0afd74…f38f01
INFO [04-28|00:12:21] 🔗block reached canonical chain       number=15202 hash
=3418a4…774312
INFO [04-28|00:12:21] 🔨mined potential block              number=15207 hash
=0afd74…f38f01
INFO [04-28|00:12:21] Commit new mining work               number=15208 txs=
0 uncles=0 elapsed=217.311µs
```

Fig.2. Transaction Submission



```
INFO [04-27|13:08:28] Commit new mining work              number=17 txs=0 uncles=0 elapsed=156.719µs
INFO [04-27|13:08:28] Successfully sealed new block       number=17 hash=542850…b11fcf
INFO [04-27|13:08:28] 🔗block reached canonical chain      number=12 hash=545f03…f3d479
INFO [04-27|13:08:28] 🔨mined potential block             number=17 hash=542850…b11fcf
INFO [04-27|13:08:28] Mining too far in the future        wait=2s
INFO [04-27|13:08:30] Commit new mining work              number=18 txs=0 uncles=0 elapsed=2.000s
INFO [04-27|13:08:31] Successfully sealed new block       number=18 hash=0154e8…4e1d94
INFO [04-27|13:08:31] 🔗block reached canonical chain      number=13 hash=b0ecf9…9575a8
INFO [04-27|13:08:31] 🔨mined potential block             number=18 hash=0154e8…4e1d94
INFO [04-27|13:08:31] Commit new mining work              number=19 txs=0 uncles=0 elapsed=113.357µs
INFO [04-27|13:08:33] Successfully sealed new block       number=19 hash=b0bbed…2de231
INFO [04-27|13:08:33] 🔗block reached canonical chain      number=14 hash=734b97…276f38
INFO [04-27|13:08:33] 🔨mined potential block             number=19 hash=b0bbed…2de231
INFO [04-27|13:08:33] Commit new mining work              number=20 txs=0 uncles=0 elapsed=139.259µs
INFO [04-27|13:08:34] Successfully sealed new block       number=20 hash=406864…df27a0
INFO [04-27|13:08:34] 🔗block reached canonical chain      number=15 hash=bbebf3…e92284
INFO [04-27|13:08:34] 🔨mined potential block             number=20 hash=406864…df27a0
INFO [04-27|13:08:34] Commit new mining work              number=21 txs=0 uncles=0 elapsed=184.75µs
INFO [04-27|13:08:35] Successfully sealed new block       number=21 hash=5e85c1…6689ce
INFO [04-27|13:08:35] 🔗block reached canonical chain      number=16 hash=9d0035…45d1db
INFO [04-27|13:08:35] 🔨mined potential block             number=21 hash=5e85c1…6689ce
INFO [04-27|13:08:35] Commit new mining work              number=22 txs=0 uncles=0 elapsed=113.337µs
INFO [04-27|13:08:36] Successfully sealed new block       number=22 hash=2b30d8…50f9ca
INFO [04-27|13:08:36] 🔗block reached canonical chain      number=17 hash=542850…b11fcf
INFO [04-27|13:08:36] 🔨mined potential block             number=22 hash=2b30d8…50f9ca
INFO [04-27|13:08:36] Commit new mining work              number=23 txs=0 uncles=0 elapsed=103.465µs
INFO [04-27|13:08:36] Successfully sealed new block       number=23 hash=4dac2e…4daa6b
INFO [04-27|13:08:36] 🔗block reached canonical chain      number=18 hash=0154e8…4e1d94
INFO [04-27|13:08:36] 🔨mined potential block             number=23 hash=4dac2e…4daa6b
INFO [04-27|13:08:36] Commit new mining work              number=24 txs=0 uncles=0 elapsed=134.174µs
INFO [04-27|13:08:38] Successfully sealed new block       number=24 hash=9858e9…32fc8d
INFO [04-27|13:08:38] 🔗block reached canonical chain      number=19 hash=b0bbed…2de231
INFO [04-27|13:08:38] 🔨mined potential block             number=24 hash=9858e9…32fc8d
INFO [04-27|13:08:38] Commit new mining work              number=25 txs=0 uncles=0 elapsed=161.303µs
INFO [04-27|13:08:38] Successfully sealed new block       number=25 hash=e0a995…140755
INFO [04-27|13:08:38] 🔗block reached canonical chain      number=20 hash=406864…df27a0
INFO [04-27|13:08:38] 🔨mined potential block             number=25 hash=e0a995…140755
INFO [04-27|13:08:38] Commit new mining work              number=26 txs=0 uncles=0 elapsed=143.896µs
INFO [04-27|13:08:38] Successfully sealed new block       number=26 hash=7495bc…d84c7a
INFO [04-27|13:08:38] 🔗block reached canonical chain      number=21 hash=5e85c1…6689ce
INFO [04-27|13:08:38] 🔨mined potential block             number=26 hash=7495bc…d84c7a
```

Fig.3. Mining done by the peers

Every user or voter is associated with a public private key. This is an add on to the security. The ‘from’ and ‘to’ addresses refer to the user or the voter and the smart contract respectively. Whenever a smart contract is deployed an address will be generated. The contract will be identified by this address throughout the network

Inorder to get a set of keys, ethereum clients such as CLI based geth, eth, pyethapp or UI based mist or chrome extensions like Metamask. In our case we have made use of geth console. An account is created it by giving a unique password. Once successful a set of public private keys will be generated. The output of account creation will be your public key. In the entire network, you will be identified by this public key. The ‘from’ address will be this public key once you perform a transaction using this address. In this way whenever a voting transaction is performed, even if the peer is compromised the vote can’t be tracked back to the voter. If you see the contents of the block mentioned above, there is no information about who is the candidate that the voter has voted for. This is an added security provided by the blockchain network. Hence in this way, security is guaranteed. Voter stays anonymous and also the candidate that the voter voted for is hidden. In this way, anonymity and transparency is maintained in the voting application.

Peers mining generate ethers as discussed. In the figure 3 if you observe you can see that most of the blocks that are getting mined do not have transactions in them. Mining is also done to increase the gas or the number of ethers in the network to fuel the running application on the network. So, empty blocks get mined all the time until a transaction is submitted so a new block having that transaction will be mined. Mining is also done to validate the voter to know if the transaction submitted by the voter is submitted by the voter himself and not an impersonation. This is also called non-repudiation and can be achieved using the concept of digital signatures. Ethereum makes use ECDSA which is Elliptical Curve Digital Signature Algorithm.

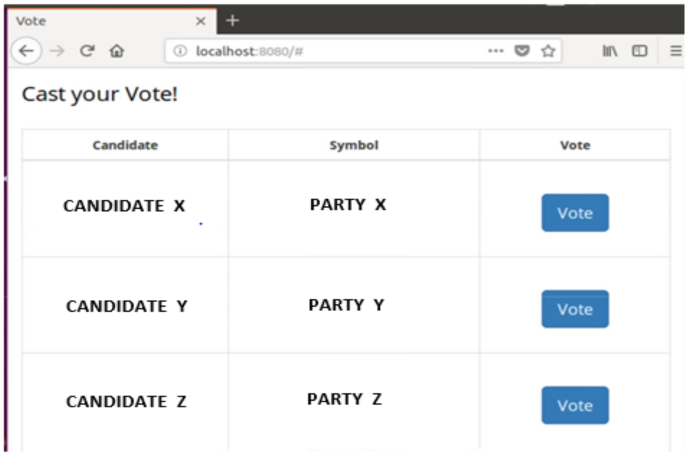
Following are the contents of Genesis block:

1. Difficulty: Measure of the toughness of mining.
2. Gas Limit: Maximum amount of gas that is spent on one transaction.
3. Gas Used: Blockchain network is brought up by using few amounts of gas/ethers.
4. Mixhash and Nonce: These terms infer if mining is completed successfully.
5. Parent Hash: This is the Keccak 256-bit hash of the previous block's header. Although it is meaningless to have this field in the genesis block since it is the first block in the blockchain, this field is used so that this block is similar to the other blocks in the blockchain.
6. Time Stamp: It is the time when the block is created as per the system time.
7. Uncle: These are the blocks that are orphaned and provides security.
8. Size: This is the size of the blockchain in bytes.

Peers are connected to each other in the private network using json RPC which makes eavesdropping impossible and tampering can't take place. All peers are synchronized and share the same blockchain. Once the smart contract is set up on this private network, the application is developed that uses this blockchain private network as backend. We have written the application using Nodejs. Nodejs along with webpack which the bundler module is to bundle the CSS, JavaScript and HTML in one file, is used as there are node modules related to web3, geth and truffles that can be used easily to create an application and deploy it on any network. In that way, Nodejs is easier. Applications can also be developed using other backend JavaScript technologies like Reactjs.

A. GUI

The user interface of the application is simplified, consisting of the candidate name and the party that the candidate belongs to. The interface doesn't allow the user to vote more than once. Once the voter finishes voting, the voter can logout of the system. The GUI in the figure 4 is redirected once the user is logged in after the authentication process via OTP. All the contents that are added to the log in, sign up page is stored in MongoDB. The contents from the voting interface is written directly to the blockchain database that is implicit to the blockchain infrastructure using JSON RPC protocol. Once the vote is casted, the transaction is submitted to the peer which is part of blockchain network. The whole process provides a level of abstraction for the complexity that happens in the backend thus giving a simplified user interface.



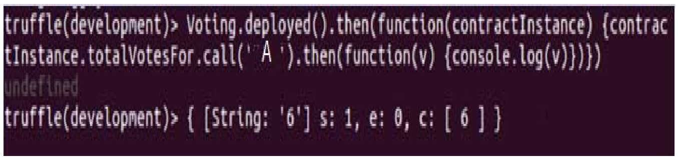
Cast your Vote!

Candidate	Symbol	Vote
CANDIDATE X	PARTY X	Vote
CANDIDATE Y	PARTY Y	Vote
CANDIDATE Z	PARTY Z	Vote

Fig.4. Voting GUI

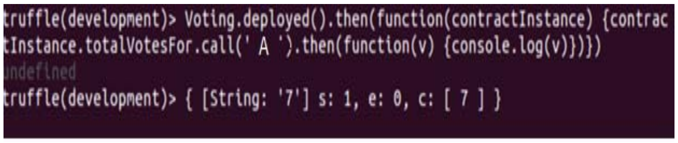
IV. Result and Discussion

Voting application has served its purpose by incrementing the vote upon casting and keeping track of the number of votes a candidate has received as you can see in the Fig.5 and Fig.6 before and after the vote is casted to candidate x.



```
truffle(development)> Voting.deployed().then(function(contractInstance) {contrac
tInstance.totalVotesFor.call(' A ').then(function(v) {console.log(v)})})
undefined
truffle(development)> { [String: '6'] s: 1, e: 0, c: [ 6 ] }
```

Fig.5. Output on querying the number of votes party A has.



```
truffle(development)> Voting.deployed().then(function(contractInstance) {contrac
tInstance.totalVotesFor.call(' A ').then(function(v) {console.log(v)})})
undefined
truffle(development)> { [String: '7'] s: 1, e: 0, c: [ 7 ] }
```

Fig.6. Output on querying the number of votes party A has after casting a vote.

In comparison to the existing methods, our proposed method is better in a way that it is simple, scalable and reliable. In the sense that it doesn't require any third-party resources like testrpc and ganache that is used to generate test ethers. In our topology we generate our own ethers by mining. The blockchain employed is private in the sense that only permissioned peers can access the blockchain. The application is easy to use and user friendly. Scalability is offered using MongoDB and the blockchain file system that is used to store the information of the blocks. Only valid users are allowed to vote and accordingly using the national identities each voter is authorized. In that way the logic is made simple and modular.

V. Conclusion

Prior to the innovation of Blockchain, electronic voting arrangements were not satisfactory on the grounds that they were not effortlessly auditable and not adequately straightforward neither for the organizers nor for voters. Furthermore, they require an expensive, work concentrated

setup. With the ethereum blockchain, any group can sort out a free, secure electronic voting. At first, groups, affiliations, or recorded companies ought to think about utilizing this innovation for government, board decisions, or general get together voting of their individuals or investors. Scaling up to national level may require coordinate contribution of the coordinators in mining activities or some level of participation with excavators. The proposed framework does not tackle every one of the issues related with electronic voting, but it provides a profitable contrasting option to present, restrictive electronic voting frameworks with the following:

1. Free, open-source peer-investigated programming
2. Ubiquitous
3. Secure
4. Permitting free, autonomous reviews of the outcomes

This app addresses security factors like verification, transparency of counting votes, integrity and non-repudiation of votes, yet it didn't address authentication of voters with strong mechanism such as biometric attributes. Ethereum and the smart contracts are the progressive achievements since the blockchain itself, precluded the restricted impression of blockchain as a digital currency (coin), and transformed it into a versatile answer for some Internet-related issues of the present world, and may empower the wide utilization of blockchain.

VI. Future Work

Generating statistics and reports based on various properties like sex, area, age and so forth. Other future work is an application more significant to the administration tasks having Aadhaar framework coordinated utilizing Aadhaar APIs. We expect that voters will utilize a safe gadget to make their choice. Indeed, even while our framework is secure, programmers can make or modify a choice utilizing vindictive programming as of now introduced on the voter's gadget. One of the disadvantages of our framework is the failure to change a vote if there should be an occurrence of a client botch. The client will have the capacity to make the choice just once. So, we might want to chip away at the issue.

VII. References

[1] Origin of Bitcoin, https://www.analyticsindiamag.com/origin-bitcoin-brief-history

[2] Overview of Blockchain Technology, "Know more about Blockchain: Overview, Technology, Application Areas and Uses Cases | #1 in Global FinTech" [Online], Available: https://medium.com/@gomedici/know-more-about-blockchain-overview technology-application-areas-and-use-cases-b48d10874293

[3] Everything you need to know about mining, https://www.bit coinmining.com.

[4] Ethereum Homestead Documentation. [Online], Available: http://www.ethdocs.org/en/latest/

[5] Ali Kaan Koç, Emre Yavuz, Towards Secure E-Voting Using Ethereum Blockchain, 1st ed, vol 1, 2018 IEEE

[6] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger", Ethereum Project Yellow Paper, vol. 151, pp. 1-32, 2014.

[7] Solidity Ethereum Virtual Machine, https://www.bitdegree.org/learn/ solidity-ethereum-virtual-machine

[8] Ethereum White Paper, https://github.com/ethereum/wiki/wiki/White-Paper

[9] E. Maaten, "Towards remote e-voting: Estonian case", Electronic Voting in Europe-Technology, Law, Politics and Society, vol. 47, pp. 83-100, 2004.

[10] N. Braun, S. F. Chancellery, and B. West. "E-Voting: Switzerland's projects and their legal framework–In a European context", Electronic Voting in Europe: Technology, Law, Politics and Society. Gesellschaft für Informatik, Bonn, pp.43-52, 2004.

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

[12] Follow My Vote: "https://followmyvote.com/online-voting-platform-benefits/open-source-code/"

[13] Moscow Government approving the idea of using blockchain for voting: "https://www.coindesk.com/blockchain-voting-code-made-open-source-moscows-government/"

[14] Ali Kaan Koç, Emre Yavuz, Towards Secure E-Voting Using Ethereum Blockchain, 1st ed, vol 1, 2018 IEEE

[15] Rinkeby Test Network: https://www.rinkeby.io/

[16] Andrew Barnes, Christopher Brake, Thomas Perry, "Digital Voting using Blockchain Technology", https://www.economist.com/sites /default/files/plymouth.com