E-Voting System -Based On Blockchain

Abstract

Democratic voting is a crucial and serious event in any country, the current voting scheme in any country is through ballot paper or by use of EVM. These processes have many drawbacks such as transparency, low voter turn-out, tampering of votes, distrust in the election body, forging of unique Id (voter id card), delay in giving out results and the most important is security issues. Security of digital voting is always the biggest concern when considering to implement a digital voting system. With such monumental decisions at stake, there can be no doubt about the system's ability to secure data and defend against potential attacks. One way the security issues can be potentially solved is through the use of blockchain technology. Electronic voting or e-voting has been used in varying forms since 1970s with fundamental benefits over paper based systems such as increased efficiency and reduced errors. However, there remain challenges to achieve wide spread adoption of such systems especially with respect to improving their resilience against potential faults.

Blockchain, the foundation of Bitcoin, has received extensive attentions recently. Blockchain serves as an immutable ledger which allows transactions take place in a decentralised manner, Security, Transparency and anonymity.Blockchain technology offers infinite number of applications. Blockchain is a distributed ledger technology that allows digital assets to be transacted in a peer-to-peer decentralised network. A distributed ledger technology is an exciting advancement in this regard. Block is a collection of all the transactions. Blockchain with smart contracts emerges as a promising candidate for building a safer, secure and transparent E-voting systems. In this paper we have implemented and tested a sample e-voting application as a smart contract for the Ethereum network using the blockchain technology through wallets and the Solidity language, that utilizes blockchain technology to securely store the casted votes and multi-factor authentication to authenticate the voters before they cast their votes while also providing an easily accessible, secure and transparent E-voting system. Limited amount of token(gas) is given in the wallet which is exhausted when the user votes thus preventing duplicity of votes. This paper also highlights the pros and cons of using blockchain technology and also demonstrates a practical system by showcasing a webapp for voting and its limitations. It presents in-depth evaluation of the scheme which successfully demonstrates its effectiveness to achieve an end-to-end verifiable e-voting scheme

Keywords— E-voting, Smart-contracts, Blockchain, Ethereum, verifiable voting, Privacy.

I. Introduction

Elections are fundamental pillar of a democratic system enabling the general public to express their views in the form of a vote. Due to their significance to our society, the election process should be transparent and reliable so as to ensure participants of its credibility. Within this context, the approach to voting has been an ever evolving domain. This evolution is primarily driven by the efforts to make the system secure, verifiable and transparent. In view of its significance, continuous efforts have been made to improve overall efficiency and resilience of the voting system. Electronic voting or e-voting has a profound role in this. Since its first use as punched-card ballots in 1960's, e-voting systems have achieved remarkable progress with its adaption using the internet technologies . However, e- voting systems must adhere to specific benchmark parameters so as to facilitate its widespread adoption. These parameters include anonymity of the voter, integrity of the vote and non-repudiation among others.

Blockchain is one of the emerging technologies with strong cryptographic foundations enabling applications to leverage these abilities to achieve resilient security solutions. Nowadays *cryptocurrency* has become a buzzword in both industry and academia. As one of the most successful cryptocurrency, Bitcoin has enjoyed a huge success with its capital market reaching 10 billion dollars in 2016 . With a specially designed data storage structure, transactions in Bitcoin network could happen without any third party and the core technology to build Bitcoin is *blockchain*, which was first proposed in 2008 and implemented in 2009 . Blockchain could be regarded as a public ledger and all committed transactions are stored in a list of blocks. This chain grows as new blocks are appended to it continuously. Asymmetric cryptography and distributed consensus algorithms have been implemented for user security and ledger consistency. The blockchain technology generally has key characteristics of decentralization, persistency, anonymity and auditability. With these traits, blockchain can greatly save the cost and improve the efficiency. A Blockchain resembles a data structure which maintains and shares all the transactions being executed through its genesis. It is primarily a distributed decentralized database that maintains a complete list of constantly germinating and growing data records secured from unauthorized manipulating, tampering and revision.

Blockchain allows every user to connect to the network, send new transactions to it, verify transactions and create new blocks. Each block is assigned a cryptographic hash (which may also be treated as a finger print of the block) that remains valid as long as the data in the block is not altered. If any changes are made in the block, the cryptographic hash would change immediately indicating the change in the data which may be due to a malicious activity. Therefore, due to its strong foundations in cryptography, blockchain has been increasingly used to mitigate against unauthorized transactions across various domains.

Bitcoin remains the most distinguished application of blockchain however researchers are keen to explore the use of blockchain technology to facilitate applications across different domains leveraging benefits such as non-repudiation, integrity and anonymity. In this paper, we explore the use of blockchain to facilitate e-voting applications with the ability to assure voter anonymity, vote integrity and end-to- verification. We believe e-voting can leverage from fundamental blockchain features such as self- cryptographic validation structure among transactions (through hashes) and public availability of distributed ledger of records. The blockchain technology can play key role in the domain of electronic voting due to inherent nature of preserving anonymity, maintaining decentralized and publicly distributed ledger of transactions across all the nodes. This makes blockchain technology very efficient to deal with the threat of utilizing a voting token more than once and the attempt to influence the transparency of the result.

The focus of our research is to investigate the key issues such as voter anonymity, vote confidentiality and end-to-end verification. These challenges form the foundation of an efficient voting system preserving the integrity of the voting process. In this paper, we present our efforts to explore the use of the blockchain technology to seek solutions to these challenges. In particular, our system is based on the Pre à Voter approach and uses an open source blockchain platform, Multichain (Multichain, 2017) as the underlying technology to develop our system. In order to protect the anonymity and integrity of a vote, the system generates strong cryptographic hash for each vote transaction based on information specific to a voter. This hash is also communicated to the voter using encrypted channels to facilitate verification.

II. BLOCKCHAIN ARCHITECTURE

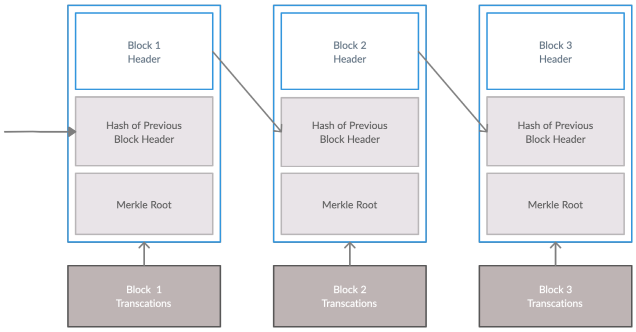

fig.1 :-An example of blockchain which consists of a continuous sequence of blocks.

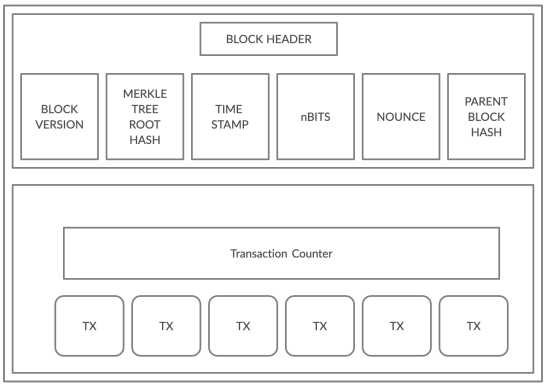

fig.2 :-Blockchain Structure

Blockchain is a sequence of blocks, which holds a complete list of transaction records like conventional public ledger Figure 1 illustrates an example of a blockchain. With a previous block hash contained in the block header, a block has only one *parent block*. It is worth noting that *uncle blocks* (children of the block's ancestors) hashes would also be stored in ethereum blockchain . The first block of a blockchain is called *genesis block* which has no parent block. We then explain the internals of blockchain in details.

A. Block

A block consists of the *block header* and the *block body* as shown in Figure 2. In particular, the block header includes: (i) Block version: indicates which set of block validation rules to follow.

(ii) Merkle tree root hash: the hash value of all the transactions in the block.

(iii) Timestamp: current time as seconds in universal time since January 1, 1970.

(iv) nBits: target threshold of a valid block hash.

(v) Nonce: an 4-byte field, which usually starts with 0 and increases for every hash calculation

(vi) Parent block hash: a 256-bit hash value that points to the previous block.

The block body is composed of a transaction counter and transactions. The maximum number of transactions that a block can contain depends on the block size and the size of each transaction. Blockchain uses an asymmetric cryptography mechanism to validate the authentication of transactions . Digital signature based on asymmetric cryptography is used in an untrustworthy environment. We next briefly illustrate digital signature.

B. Digital Signature

Each user owns a pair of private key and public key. The private key that shall be kept in confidentiality is used to sign the transactions. The digital signed transactions are broadcasted throughout the whole network. The typical digital signature is involved with two phases: *signing phase* and *verification phase*. For instance, an user Alice wants to send another user Bob a message. In the signing phase, Alice encrypts her data with her private key and sends Bob the encrypted result and original data. In the verification phase, Bob validates the value with Alice's public key. In that way, Bob could easily check if the data has been tampered or not. The typical digital signature algorithm used in blockchains is the elliptic curve digital signature algorithm (ECDSA).

C. Key Characteristics of Blockchain

In summary, blockchain has following key characteristics.

• *Decentralization* :In conventional centralized transaction systems, each transaction needs to be validated through the central trusted agency (e.g., the central bank), inevitably resulting to the cost and the performance bottle- necks at the central servers. Contrast to the centralized mode, third party is no longer needed in blockchain. Consensus algorithms in blockchain are used to maintain data consistency in distributed network.

• *Persistency*. Transactions can be validated quickly and invalid transactions would not be admitted by honest miners. It is nearly impossible to delete or rollback transactions once they are included in the blockchain. Blocks that contain invalid transactions could be discovered immediately.

- *Anonymity:* Each user can interact with the blockchain with a generated address, which does not reveal the real identity of the user. Note that blockchain cannot guarantee the perfect privacy preservation due to the intrinsic constraint .

- *Auditability:* Bitcoin blockchain stores data about user balances based on the Unspent Transaction Output (UTX-O) model: Any transaction has to refer to some previous unspent transactions. Once the current transaction is recorded into the blockchain, the state of those referred unspent transactions switch from unspent to spent. So transactions could be easily verified and tracked.

E-VOTING BACKGROUND AND REQUIREMENTS

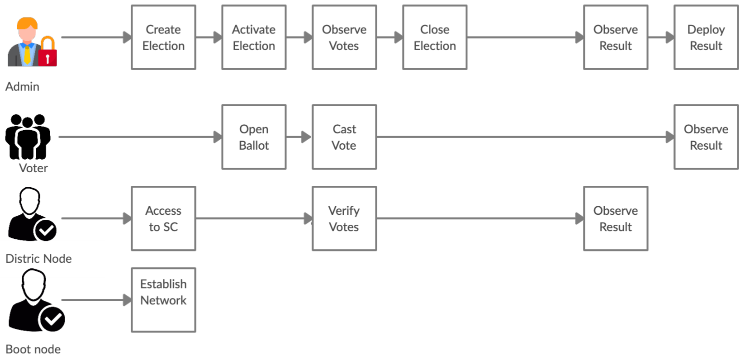

Fig.3. Election roles and process

Electronic voting has been an area of research focus for many years by using computing machines and equipment for casting votes and producing high quality and precise results in accordance with the sentiments of the participating voters. Various attempts have been adopted in practice to support election process. Initially computer counting system allowed the voter to cast vote on papers. Later on, those cards went through the process of scanning and tallying at every polling cell on a central server. Direct Recording Electronic (DRE) voting systems were put in place later on which were admired and acknowledged greatly by the voters in-spite of the resistance from computer scientists. If the voting system is well understood by the voters, the system's usability can be increased remarkably. DRE systems in particular have gathered a lot of successes in bringing the voters to use this technology. These systems work more or less in the same way as any conventional election system does. In the case of DRE, a voter begins his journey by going to their polling place and get their token to vote where he utilizes his token at the voting terminal to vote for his candidate. When the candidate selection procedure is completed, DRE systems present the final selection to the voter before actually casting it (in case if the voter wants to change his opinion) and after the final selection, the ballot casting is completed

More recently, distributed ledger technologies such as blockchain have been used to achieve e-voting systems primarily due to their advantages in terms of end-to-end verifiability. With properties such as anonymity, privacy protection and non-repudiation, blockchain is a very attractive alternative to contemporary e-voting systems. The research presented in this paper also attempts to leverage these properties of blockchain to achieve an efficient e-voting system. A detailed analysis of such systems is presented in the next section along with the identification of comparison with these approaches.

I present a brief description of each requirement along with an explanation of how the proposed system fulfils it.

Privacy - Keeping an individual's vote secret

The system leverages cryptographic properties of blockchain to achieve privacy of a voter. More specifically, as voter is registered into the system, a voter hash is generated by blockchain which is the unique identifier of a voter into the blockchain, and is protected from misuse due to collision resistance property of the cryptographic hash. Due to this, the traceability of a vote is also non-trivial thereby protecting the voter when under duress.

Eligibility - Allowing only registered voters to vote, with each such voter voting only once

All eligible users are required to register using unique identifiers such as government-issued documents to assert their eligibility. In addition to this, our system implements strong authentication mechanism using finger printing technology to assert that only authorized voters can access the system. Furthermore, the use of biometrics also enables the system to protect against double voting.

Receipt Freeness - Voters should be unable to prove to a third party that they voted in a particular way

The proposed system enables a voter to vote as per their choice and creates a cryptographic hash for each such event (transaction). This is important to achieve verifiability i.e. to verify if a certain vote was included in the count. However, possession of this hash does not allow to extract information about the way voter has voted.

Convenience - Voters must be able to vote easily, and everyone who is eligible must be able to vote

The system has been implemented using a user friendly web based interface with the voting process requiring minimal input from the user. For instance, fingerprinting is implemented for authentication mechanism to avoid the requirement to remember username/passwords. Furthermore, the overall process is integrated which enables the user to interact with it in a seamless manner.

Verifiability - The ability to trust the vote tallying process upon casting their vote successfully, a user is provided with their unique transaction ID in the form of a cryptographic hash. A user can use this transaction ID to track if their vote was included in the tallying process. However, this process does not enable a user to view how they voted which has been adopted to mitigate threats when under duress.

Eligibility - To casting their vote successfully, the user has to verify their eligibility to cast their vote or not. First and foremost the user should provide their unique identification (voter ID , Aadhar) to verify their election eligibility by checking their age whether voters age is 18+ or not, this process which has been adopted to decrease threats from non-eligible voters.

The analysis presented above highlights the performance of the proposed system with respect to the specific requirements of e-voting. It also highlights the significance of defining characteristics of blockchain and their profound role in achieving the cornerstones of an efficient e-voting system. Therefore, we believe the work presented here makes significant contribution to the existing knowledge with respect to the application of blockchain technology to achieve a secure digital voting system.

III. RELATED WORKS

Our main motivation in this project is to provide a secure voting environment and show that a reliable e-voting scheme is possible using blockchain. Because, when e- voting is available for everyone who has a computer, or a mobile phone, every single administrative decision can be made by people and members; or at least people's opinion will be more public and more accessible by politicians and managers. This will eventually lead humanity to the true direct democracy . It's important for us since elections can easily be corrupted or manipulated especially in small towns, and even in bigger cities located in corrupt countries. Plus, large-scale traditional elections are very expensive in the long term, especially if there are hundreds of geographically distributed vote centers and millions of voters. Also, the voter turn-out at the voting centers is relatively low as the person might not be staying at the address his name is enrolled in the list, or he might be out for vacation or any other work. E-voting will be able solve these problems, if implemented carefully. The concept of e- voting is significantly older than blockchain. So that, all known examples so far used means of centralized computation and storage models.

Estonia is a very good example, since the government of Estonia is one of the first to implement a fully online and comprehensive e-voting solution. The concept of e- voting was started to be debated in the country in 2001 and officially started by the national authorities in the summer of 2003. Their system is still in use, with many improvements and modifications on the original scheme. As reported, it is currently very robust and reliable. They use smart digital ID cards and personal card readers (distributed by the government) for person-wise authentication. For citizens to attend the elections by listing the candidates and casting a vote, there is a special web portal as well as an equivalent desktop app. So that, anyone having a computer and Internet connection and also his/her ID card, can easily vote remotely.

People can also digitally create petitions and proposals for acts and laws at the parliament's website (http://rahvaalgatus.ee). These petitions can be digitally signed using the smart ID card by any citizen who wants to support the proposal. If proposals achieve a certain number of signatures, they are discussed in the parliament. That's another good example showing how technology can strengthen the democracy. Though being considerably successful and reaching nearly 30% penetration rate during recent elections, the Estonian model has some drawbacks, too. The centralized solution, by its nature, creates a single- point-of-failure and is open to hacking/hijacking attempts. In example, Distributed Denial of Service (DDoS) attacks can harm the software, servers or databases used. The administrators of such a system may act malicious and steal, if cannot manipulate, some valuable information during an election. The scalability of this system is another question. Since Estonia has a relatively small population, it is hard to estimate if such a system would work flawlessly in, say, China. The constant need for the ID card and the reader device is not nice, too, due to the extra cost of producing, distributing, and carrying (for voters) them.

Switzerland is another one of the few countries participating in the electronic voting trend. In Switzerland, known for its widespread democracy, every citizen who completes the age of 18 can take an active or passive role in the elections, which may be held in many different topics for many different decisions. They have also begun an official work on a voting system called remote voting . Example - Sierra Leone's March 2018 general election – Swiss startup Agora carried out tallying in two districts. After the voting, a team of accredited observers from different locations manually entered approximately 400,000 ballots into Agora's blockchain system. These system was the partial deployment of blockchain and the votes were verified by blockchajn and they were not blockchain powered

similar commercial or experimental work was carrried out in the Russian city of Moscow for its Active Citizen program –In December 2017. the program started using a blockchain for voting and to make the voting results publicly auditable. Each question discussed by the community and put up for voting is moved to the e-voting system using a blockchain. After the voting is complete, the results are listed on a ledger containing all the previous poll

As an online polling example, rather than an e-voting system, http://www.strawpoll.me/ is a popular and free service. It's a simple website that allows everyone to create questionnaires and allows answering others' polls with votes. It shows how powerful can be e-voting, because everyone easily accesses the election and uses his/her votes and declares his/her choice. People can share private hyperlinks to any created poll (as long as they know the link) and people who have the link can vote and one browser can only use one vote. The security here, in terms of voter authentication, duplicate votes and non-repudiation of votes, is very weak. http://www.strawpoll.me/ trusts people about they will not violate the election process while benefiting ease of access and using features of e-voting. Hence, it cannot be used in real cases such as choosing the chairman of a department, etc

IV. PRELIMINARIES

- Blockchain Mining

To reach consensus on the state of the blockchain in a trust-less network, a concept known as 'mining' is employed. The role of a miner node is to verify transactions, group transactions into blocks, and append them to the blockchain. To append a new block to the blockchain, the hash of the block must begin with a certain number of zeros. To achieve this, a number called a 'nonce' is included in each block; each time miners hash the block without solving the computational problem, they increment the nonce and rehash the block. The difficulty of solving the hashing problem is described as 'Proof of Work,' signifying the computational power and difficulty needed to append a new block to the blockchain . Be- cause of the computational power needed to mine the blockchain, miners are rewarded: for instance, in Bit- Coin, when miners successfully append new blocks to the blockchain, they are rewarded with the current payout rate in bitcoins.

- Eth Calls

Every valid transaction executed is stored on the blockchain. Due to this, blockchains can suffer from scalability issues. Valid transactions sent to smart contracts in the Ethereum blockchain are considered state changeable calls and consume gas. To reduce gas consumption and the number of transactions on the blockchain, the Ethereum blockchain allows *eth.calls* to be utilized in addition to transactions. *Eth.calls* allow nodes to send messages to other nodes or smart contracts to retrieve its current state without storing the message on the blockchain .Therefore, *eth.calls* are similar to simulations of transactions. By executing *eth.calls* to send notifications/messages or to retrieve current states, the size of the blockchain can be greatly reduced.

- Paillier Encryption

Full homomorphic encryption enables users to per- form computations on encrypted data that can be decrypted and yield the same result as if the computation had been originally performed on decrypted data. However, doing fully modular multiplication in fully homomorphic encryption is computationally intensive and very slow. Nonetheless, because of the advantages of homomorphic encryption, partial homomorphic encryption is a prominent encryption scheme. One such scheme is Paillier Encryption. This probabilistic public-key encryption method supports addition and multiplication. Paillier system can homomorphically add two cipher texts but it can only multiply a cipher text with a plaintext integer. Since the Paillier system cannot homomorphically multiply two cipher texts, it is considered partially homomorphic. The process of encryption is not completely intuitive: multiplying cipher texts is equivalent to adding the plain texts and raising a cipher text to the power of another cipher text is equivalent to multiplying the plain texts. To achieve the advantages of homomorphic encryption without the substantial reduction in processing speed, Paillier Encryption is one of the ideal encryption schemes.

- MetaMask

MetaMask was created to increase the accessibility of the Ethereum blockchain to the average user. A plug-in for Chrome, MetaMask acts as an Ethereum browser, allowing users to manage their Ethereum wallet and interact with decentralized applications and smart contracts without running a full node. Through MetaMask, users are able to manage multiple accounts and easily switch between different networks . In order to allow users the flexibility of using the Ethereum blockchain without running a full node, MetaMask relies on trusted nodes to broadcast the transactions of MetaMask users in order to be mined. Since transactions are signed us- ing the sender's private key, which is stored locally on the user's machine, MetaMask cannot impersonate the user and send transactions on the user's be- half. Acting as an intermediary between Chrome and the Ethereum blockchain, MetaMask allows users the convenience and security of the blockchain within a popular browser.

V. IMPLEMENTATION AND DISCUSSION

In this section we will illustrate the design and functional phase of our application, The User accesses the web application where the platform is hosted and register's itself as well as cast its vote in an secured and transparent manner. Fig 3 depicts the overview of the application.

1. Registration Phase: The Voter has to Register itself first with its unique id and attributes such as name roll no and mobile number. All this data is stored in the database.

2. Login: The voter after registration tries to login themselves to cast a vote. In this phase voter first logs in using password. After successful login, to cast their vote voter has to authenticate themselves. For real-time authentication OTP verification is used for enhanced security.

3. Blockchain Technology: This technology is mainly used for its security features. Blockchain provides a secure and transparent environment. Blockchain encrypts the voter message (Casted vote) using Asymmetric encrypting algorithm. A public key is provided by Blockchain and private key is with host. Public key is used for verification purpose by ledger..

4. Database: User database is stored in database. Details like name, gender, Unique Id are stored is database. MySQL is the proposed database to be used.

5. Ethereum Network: Ethereum network provides a framework for blockchain creation and storage. Every block is created and its details are stored in an encrypted ledger. These created blocks are distributed among nodes which provides high fault tolerance to the system.

6. Result phase: The processing and tallying of votes is done in results phase. Results are generated and displayed on website. Users can verify their votes using their own public key. This provides transparency to the voting system

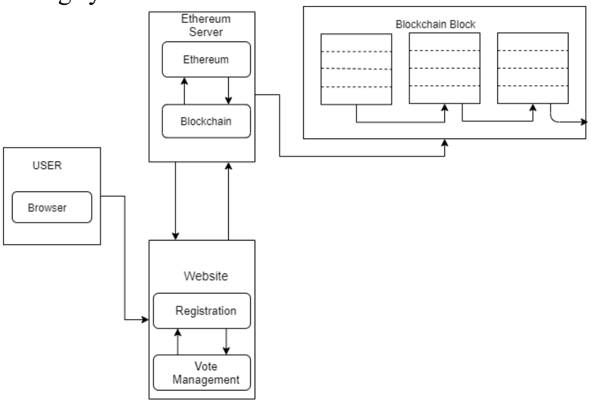

Fig. 4.Whole System Overview

The application is built using the architectural pattern of Model-View-Controller. It is also widely used architecture. Here, the application is divided into three main logical components: the model, the view and the controller.

- View: The top layer is where the end-user communicates with the application through clicking buttons, typing details, accessing camera, selecting radio button, uploading songs, etc. This layer is responsible for displaying all data or a portion of data to user based on the requirement of the application. This layer also acts as a bridge between the user and application itself.

- Controller: This middle layer of the application contains the business logic, and the main functionality of the application. As soon as the user interacts with the application, the response is processed in this layer. From log-in to casting vote, all the functions that run in background belong to this layer. This mainly consists of all the functions and sending output to view layer

- Model: This layer is responsible for maintaining the user's data. Relational Database MySQL is used for storing user data.

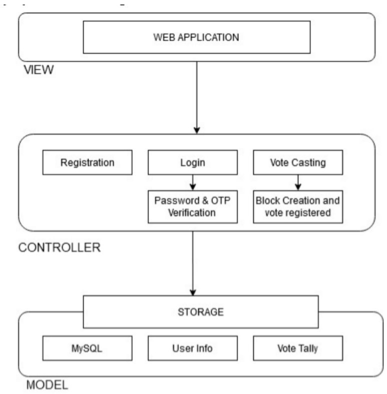

Fig. 5.MVC(Model-View-Controller) architecture

In our application for a user to vote he/she needs an account with a wallet address and some Ether, Ethereum's cryptocurrency. After connecting to the network they cast their vote and pay a small transaction fee to write their vote to the blockchain. This transaction fee is called as "gas" in our application which can be related to some coins. This transaction fee "gas" is awarded to the miner-node of the network after he completes the transaction. It's important to note that voting on the blockchain costs us some Ether but seeing the list of candidates is free, because writing to blockchain costs but reading data from the blockchain is free.

To code our application Ethereum blockchain allows us to execute code with Ethereum Virtual machine (EVM) on blockchain with smart contract. In our application Smart contracts are responsible of reading and writing data to the blockchain as well as executing the logic. Smart contracts are written in programming language called Solidity. If the public ledger represents database layer of the blockchain, then smart contracts are where all the business logic that transacts with that data lives. Smart contracts represent a covenant or agreement, In our application its is an agreement that user's vote will count, others vote will be counted only once and the candidates with highest vote will be declared the winner.

Step first to build our application is installing all the dependencies and then writing our contract and deploying it to the blockchain successfully. To create the contract

```
       contract Election {
         // Model a Candidate
         struct Candidate {
          uint id;
          string name;
          uint voteCount;
          }
      constructor () public {
       addCandidate("candidate 1 ");
       addCandidate("candidate 2 ");
       }
}
```

Fig 6 – Code block to define struct variable and contract

We have specified that struct candidate has an id of unsigned integer type, name of string type, and the vote count of unsigned integer type. To store these structs we use solidity mapping which is like associative array or a hash, that associates key-value pairs.

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

here the key to mapping is unsigned integer and value is Candidate structure type and mapping's visibility is set to public so as to get a getter function.

The complete contract code contains mapping, function to add candidates and smart contract called contract election .

```
contract Election
// Model a Candidate
 struct Candidate {
               uint id;
               string name;
               uint voteCount;
           }
// Read/write candidates mapping(uint => Candidate) public
candidates;

          // Store Candidates Count
          uint public candidatesCount;
function Election () public { addCandidate("Candidate 1");
addCandidate("Candidate 2");

}

function addCandidate (string _name) private {

            candidatesCount ++;
            candidates[candidatesCount]
            Candidate(candidatesCount,
_name, 0); }

}
```

Fig. 7. Code block of complete contract code

After creating the server side application we created client side application that will talk to our smart contract. we created our front-end with java script and HTML. To make our system more secure we have included one more unique feature other than unique id and password is the OTP(one time password) feature. We request users to enter their mobile no on which otp is send and then the system verifies the user.

After creating the webpage we need to log in to the blockchain. To connect to blockchain we need to import one of the accounts from ganache- One of the dependencies which gives us 10 accounts with account address and some fake ethers, into MetaMask. In order to use blockchain we must install a special browser extension in order to use the Ethereum blockchain. That's where MetaMask is used. After connecting we can interact with our smart contract and will be able to see our contract and account data.

The next step was to add the ability to cast votes in the elections. To keep the track of accounts that have voted we define voters and mapped it to the smart contract, and add 'vote' function which takes in one argument- candidate-Id. It checks that the user hasn't voted before, candidate is valid, recording that user has voted after his voting and then update the candidate vote count. Fig-8 depicts the code and mapping for casting the vote

```
// Store accounts that have voted mapping(address => bool) public voters;

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

// require a valid candidate

require(_candidateId > 0 && _candidateId <= candidatesCount);

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

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

// trigger voted event

emit votedEvent(_candidateId); }
```

Fig 8 – Code Bock for casting of vote/ vote process

when the user votes by using gas which is rewarded to the node(miner) whoever writes it to the blockchain, after successful casting of votes results are displayed and candidate with highest votes is the winner.
In this project, our scope is limited for small-scale polls and elections such as college elections. A larger voting with millions of voters may have different problems to address. The Ethereum network's scalability is still unknown and needs further research, that's why we cannot suggest use of these contracts for nation-wide elections, at least for now. Our contracts are executed in the Ethereum blockchain, so wherever browser can be run (location, platform, device, etc.), our voting application can be used, too. A fundamental problem of blockchain based e-voting systems is to provide anonymity for voters without compromising the transparency of the general voting process. In detail, all the transactions (money transfers, votes etc.) are essentially written to the blocks of the blockchain as plaintext. So that, a vote from wallet address A to wallet address B can be seen by anyone who has access to the chain. Which is, of course, a big disadvantage. And, it is not possible to use such a system for official/critical elections. Providing this anonymity is also a major challenge in the current state-of- the-art works.

VI. CONCLUSION AND FUTURE WORK

In this paper, we introduced a unique, blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost-efficient election while guaranteeing voters privacy By comparison to previous work, we have shown that the blockchain technology offers a new possibility for democratic countries to advance from the pen and paper election scheme, to a more cost- and time- efficient election scheme, while increasing the security measures of the todays scheme and offer new possibilities of transparency.

Electronic voting has been used in varying forms since 1970s with fundamental benefits over paper based systems such as increased efficiency and reduced errors. With the extraordinary growth in the use of blockchain technologies, a number of initiatives have been made to explore the feasibility of using blockchain to aid an effective solution to e-voting. This paper has presented one such effort which leverages benefits of blockchain such as cryptographic foundations and transparency to achieve an effective solution to e-voting. The proposed approach has been implemented with Multichain and indepth evaluation of approach highlights its effectiveness with respect to achieving fundamental requirements for an e-voting scheme.

E-voting is still a controversial topic within both political and scientific circles. Despite the existence of a few very good examples, most of which are still in use; many more attempts were either failed to provide the security and privacy features of a traditional election or have serious usability and scalability issues . On the contrary, blockchain-based e-voting solutions, including the one we have implemented using the smart contracts and the Ethereum network, address (or may address with relevant modifications) almost all of the security concerns, like privacy of voters, integrity, verification and non-repudiation of votes, and transparency of counting. Yet, there are also some properties that cannot be addressed solely using the blockchain, for example authentication of voters (on the personal level, not on the account level) requires additional mechanisms to be integrated, such as use of biometric factors. In continuation of this work, we are focused at improving the resistance of blockchain technology to 'double spending' problem which will translate as 'double voting' for e-voting systems. Although blockchain technology achieves significant success in detection of malleable change in a transaction however successful demonstration of such events have been achieve which motivates us to investigate it further.

Blockchain technology has lot of promise, but in its current state its require lot more research and currently might not reach till its full potential. There needs a concerted effort in the core blockchain technology to improve its support for more complex applications. To this end, we believe an effective model to establish trustworthy provenance for e-voting systems will be crucial to achieve an end-to-end verifiable e-voting scheme. The work to achieve this is underway in the form of an additional provenance layer to aid the existing blockchain based infrastructure.

REFERENCES

[1] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system”, [Online]. Available: https://bitcoin.org/bitcoin.pdf .

[2] Ali Kaan Koç, Emre Yavuz, Umut Can Çabuk, Gökhan Dalkılıç “Towards Secure E-Voting Using Ethereum Blockchain”

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

[4] C.D. Clack, V.A. Bakshi, and L. Braine, “Smart contract templates: foundations, design landscape and research directions”, Mar 2017, arXiv:1608.00771

[5] Adida, B.; ‘Helios (2008). Web-based open-audit voting, in Proceedings of the 17th Conference on Security Symposium, ser. SS'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 335{348.

[6] Adida B. and Rivest, R. L. (2006). Scratch & vote: Self-contained paper-based cryptographic voting, in Proceedings of the 5th ACM Workshop on Privacy in Electronic Society, ser. WPES '06. New York, NY, USA: ACM, 2006, pp. 29-40.

[7] Bell, S., Benaloh, J., Byrne, M. D., Debeauvoir, D., Eakin, B., Kortum, P., McBurnett, N., Pereira, O., Stark, P. B., Wallach, D. S., Fisher, G., Montoya, J., Parker, M. and Winn, M. (2013). Star-vote: A secure, transparent, auditable, and reliable voting system, in 2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13). Washington, D.C.: USENIX Association, 2013.

[8] Bohli, J. M., Muller-Quade, J. and Rohrich, S. (2007). Bingo voting: Secure and coercion- free voting using a trusted random number generator, in Proceedings of the 1st International Conference on E-voting and Identity, ser. VOTE-ID'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 111-124.

[9] 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.

[10] NirKshetri,JeffreyVoas,“Blockchain-EnabledE-Voting”.

[11] P. McCorry, S.F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy", International Conference on Financial Cryptography and Data Security. Springer, Cham, pp. 357-375, 2017.

[12] U.C. Çabuk, T. Şenocak, E. Demir, and A. Çavdar, “A Proposal on initial remote user enrollment for IVR-based voice authentication systems”, Int. J. of Advanced Research in Computer and Communication Engineering, vol 6, pp.118-123, July 2017.

[13] Y.Takabatake,D.Kotani,andY.Okabe,“Ananonymousdistributed electronic voting system using Zerocoin“, IEICE Technical Report, pp. 127-131, 2016.