Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

E-Voting using Blockchain Technology

Abhishek Subhash Yadav
Computer Engineering Dept.
MIT College of Engineering, Pune

Yash Vandesh Urade
Computer Engineering Dept.
MIT College of Engineering, Pune

Ashish Uttamrao Thombare
Computer Engineering Dept.
MIT College of Engineering, Pune

Abhijeet Anil Patil
Computer Engineering Dept.
MIT College of Engineering, Pune

***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. 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 decentralized network. A distributed ledger technology is an exciting advancement in this regard. Block is a collection of all the transactions. Blockchain possess salient features such as immutability, Decentralization, Security, Transparency and anonymity. 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. 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.**

Keywords— E-voting, Smart-contracts, Blockchain, Ethereum

I. INTRODUCTION

Blockchain technology that shines like a star after the entrance and widespread acceptance of Bitcoin [1], the very first cryptocurrency in peoples' everyday life, has become a trending topic in today's software world[2]. Blockchain technology originates from the underlying architectural design of the cryptocurrency bitcoin, where it was first introduced to the internet world and sooner became a promising technology due its high degree of transparency in the system it become an active field of research and study for its application various other fields For example, in Bitcoin, since the wallets are in a distributed structure, the total amount of coins and instant transaction volume in the world can be followed momentarily and clearly. There is no need for a central authority to approve or complete the operations on this P2P-based system.

Because of that, not only the money transfers but also all kinds of structural information can be kept in this distributed chain, and with the help of some crypto-logical methods, the system can be maintained securely. Like people's assets, marriage certificates, bank account books, medical information, etc., a lot of information can be recorded with this system with relevant modifications [3]. Ethereum coin (Ether), another cryptocurrency with multipurpose development environments, which emerged a few years after Bitcoin, distinguishes the blockchain in a real sense, revealing that this technology can produce software that can hold information that is structured as described above. The software programs enforced by smart contracts [4] (explained later) are written into the blockchain and are immutable. They cannot be (illegally) removed nor manipulated once written. Hence, they can work properly, autonomously and transparently forever, without any external stimuli [5].

Blockchain stores transactions in a block, the block eventually becomes completed as more transactions are carried out. Once complete it is then added in a linear, chronological order to the blockchain

The initial block in a blockchain is known as the 'Genesis block' or 'Block 0'. The genesis block is usually hardcoded into the software; it is special in that it doesn't contain a reference to a previous block. ('Genesis Block', 2015) Once the genesis block has been initialised 'Block 1' is created and when complete is attached to the genesis block. Each block has a transaction data part, copies of each transaction are hashed, and then the hashes are paired and hashed again, this continues until a single hash remains; also known as a merkle root (Figure 1).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

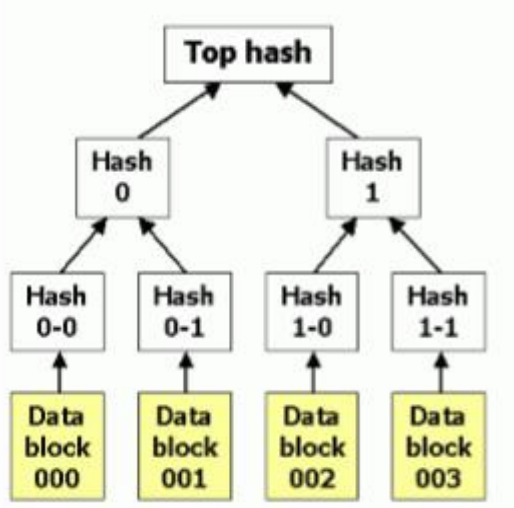

Figure 1- Hash table

The block header is where the merkle root is stored. To ensure that a transaction cannot be modified each block also keeps a record of the previous blocks header, this means to change data you would have to A blockchain is designed to be accessed across a peer-to-peer network, each node/peer then communicates with other nodes for block and transaction exchange. Once connected to the network, peers start sending messages about other peers on the network, this creates a decentralised method of peer discovery. The purpose of the nodes within the network is to validate unconfirmed transactions and recently mined blocks, before a new node can start to do this it first has to carry out an initial block download. The initial block download makes the new node download and validate all blocks from block 1 to the most current blockchain, once this is done the node is considered synchronised

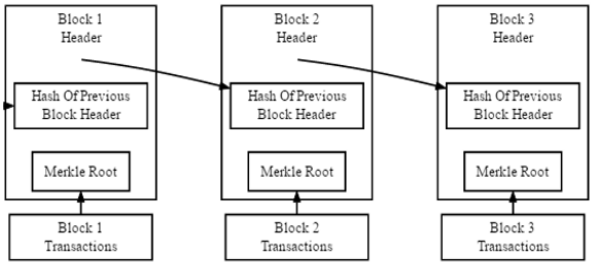

Figure-2 Simplified bitcoin blockchain(source-bitcoin.org)

Blockchain might be a suitable solution for e-voting projects. E-voting is being studied extensively, and many implementations are tested and even used for a while. However, very few implementations are reliable enough and are still in use. Of course, there are many successful examples of online polls and questionnaires, yet we cannot claim the same for online elections for governments and businesses. That's mainly because, official elections are essential elements of the democracy and democratic administrations, which are the most preferred administrative methodology in the modern world. More, what is most valued in democratic societies is a robust electoral process that provides transparency and privacy. Today, a lot of decisions are being made by people (and members in organizations)[2].

The current voting scheme raises many questions such as how reliable and transparent the system is, are the votes not changed before they are counted, how can we verify the transparency of the system. So to tackle this kind of questions in this paper we investigate and propose a web-application using the blockchain technology over the Ethereum server by deploying smart contracts. In the 2 section we discuss about the current deployment of e-voting systems and further on we discuss limitations of it.

II. MOTIVATION AND 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 [6]. 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 [7]. 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 [8]. 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 [9]. 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 [10]. 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.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

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 [11]. 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[10]

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[2].

In this paper we build a system by integrate the blockchain paradigm into e-voting procedure and come up with a feasible and general e-voting protocol without a TTP, which provides a secure and flexible voting mechanism, which satisfies almost all of the main requirements for an e-voting system and strengthens the power of the election organized.

III. 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

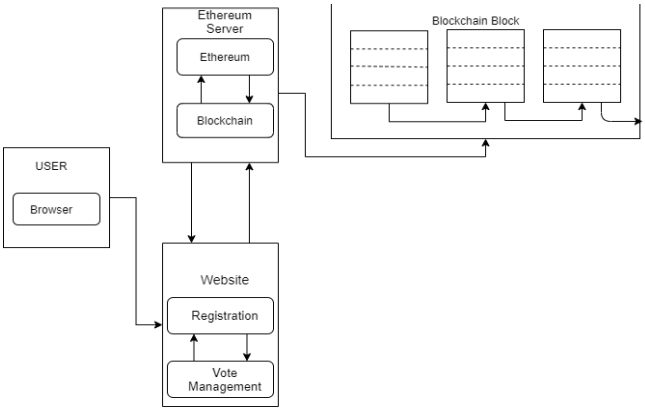

Figure 3 – System Overview

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

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.

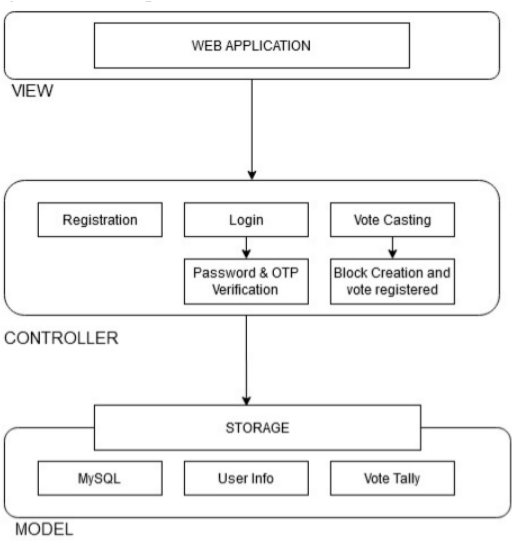

Figure-4 MVC architecture

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 declare the smart contract with the "contract" keyword, followed by the contract name. Next, we declare a state variable that will store the value of the candidate name. State variables allow us to write data to the blockchain constructor is called whenever we deploy the contract. Fig – 5 shows the structure, variable and contract declaration.

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

Fig 5 – 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");
}
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

```
function addCandidate (string
_name) private {
        candidatesCount ++;
        candidates[candidatesCount]
        Candidate(candidatesCount,
_name, 0);
        }
        }
```

Fig. 6. 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.

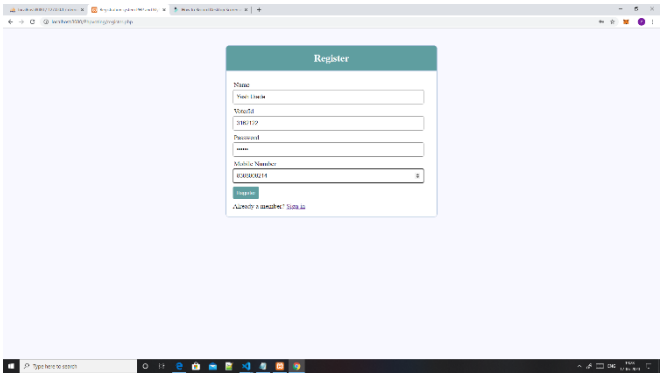

Fig 8 – Screenshot of application during registration process

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-9 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 9 – 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.

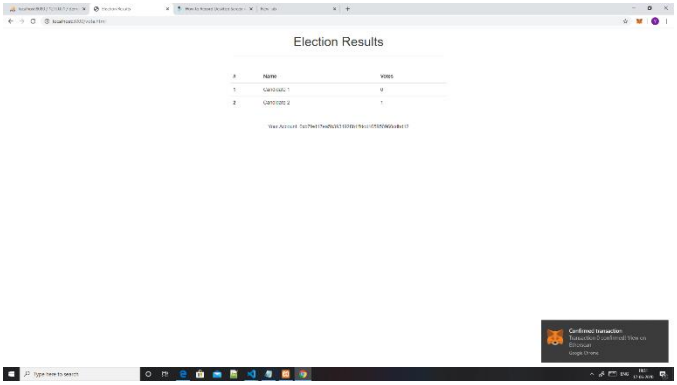

Fig. 10. Screenshot of a Election results

```
eth_getFilterChanges
eth_sendTransaction

  Transaction: 0x1ec4dbbbde77ab2e5e65727e2d7f0a62ed5f43bf9ba78046908862
aa4492e2cf
  Gas usage: 66241
  Block Number: 5
  Block Time: Mon Jul 13 2020 22:19:22 GMT+0530 (India Standard Time)

eth_getTransactionReceipt
eth_getFilterChanges
eth_coinbase
```

Fig. 11. Screenshot of a vote casting entry in the chain.

Fig 10 shows the elections results after successful casting of vote and fig 11 shows vote casting entry in the chain with transaction hash, blocks-created till now, contract address, timestamp, account, block number of the transaction, gas used, and the total cost during the whole process of casting.

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,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

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. Hao et al. in their work, proposed a solution based on the Diffie-Hellman process, which also implies the use of public/private key pairs and random numbers, so that a "two-round" referendum can supposedly be held with some ballot privacy [12]

IV. CONCLUSION

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.

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 [8]. 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 [13].

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.

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] E. Maaten, "Towards remote e-voting: Estonian case", Electronic Voting in Europe-Technology, Law, Politics and Society, vol. 47, pp. 83-100, 2004.
[6] U.C. Çabuk, A. Çavdar, and E. Demir, "E-Demokrasi: Yeni Nesil Doğrudan Demokrasi ve Türkiye'deki Uygulanabilirliği", [Online] Available: https://www.researchgate.net/profile/Umut_Cabuk/publication/308796230_E-Democracy_The_Next_Generation_Direct_Democracy_and_Applicability_in_Turkey/links/5818a6d408a ee7cdc685b40b/E-Democracy-The-Next-Generation-DirectDemocracy-and-Applicability-in-Turkey.pdf.
[7] "Final report: study on eGovernment and the reduction of administrative burden (SMART 2012/0061)", 2014, [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/finalreport-study-egovernment-and-reduction-administrative-burdensmart-20120061
[8] F. Hao and P.Y.A. Ryan, Real-World Electronic Voting: Design, Analysis and Deployment, CRC Press, pp. 143-170, 2017.
[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] Nir Kshetri, Jeffrey Voas, "Blockchain-Enabled E-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, and Y. Okabe, "An anonymous distributed electronic voting system using Zerocoin", IEICE Technical Report, pp. 127-131, 2016.