INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 5)
Available online at: www.ijariit.com

Design and implementation of a secure and robust voting system based on blockchain

Ankush Pathak
ankushvpathak@acm.org
Pimpri Chinchwad College of Engineering, Pune, Maharashtra

Abdul Wasay
wasaya@acm.org
Pimpri Chinchwad College of Engineering, Pune, Maharashtra

Chandan Singh
sc231997@gmail.com
Pimpri Chinchwad College of Engineering, Pune, Maharashtra

Ritvik Bhavan
ritvik.bhavan@gmail.com
Pimpri Chinchwad College of Engineering, Pune, Maharashtra

Dr. Jayant Umale
jayantumale@gmail.com
Pimpri Chinchwad College of Engineering, Pune, Maharashtra

ABSTRACT

In recent times, numerous accusations have been raised on the integrity of the EVMs used in the Indian elections. These accusations demand the need of a newer system that is in line with the present modern era and addresses these accusations. This new system must be an amalgamation of technology and trust, of modernity and tradition. It must be secure, auditable, and transparent and should reinforce the confidence of the voters in the democratic election process. Blockchain technology allows for development of a decentralized distributed open ledger. It offers features like immutability of data, integrity of data and resistance of data to modifications. We aim to use these features of Blockchain to build a voting machine that will solve existing issues with EVMs and will be sufficiently automated.

Keywords— *Blockchain, EVM, Node, Network, Ethereum, Immutability, Voting*

1. INTRODUCTION

1.1 Voting

In a democracy, the supreme power is vested in the people of the country and is exercised directly or indirectly by them through their chosen representatives. [1]. The representatives are chosen through a system of periodically held system of elections. After an election, the majority party or coalition forms the government and its leader becomes the Prime Minister.

Thus, it's quite natural and logical to conclude that the voting process is pivotal to the normal functioning of a democracy. The core working principle of a democracy, that is of a government by the people, of the people and for the people is established through the right to vote. This right is given to every functioning member of the democracy. If the right to vote was compromised, it would be in direct violation of the core working principle of a democracy.

In India, the voting process was conducted through a practice of secret ballot. In 1998, the process of secret ballots gave way to the use of Electronic Voting Machines (EVMs). In the 2004 Lok Sabha elections the Election Commission of India (ECI) took a historic decision to use only EVMs. Since then India has never looked back onto secret ballots [2]. The secrecy of the vote becomes one of the fundamental principles required to successfully conduct democratic elections. [3] Failure to secure this leads to violation of the free will of the voter. Undermining their right towards expression of free will. [3] In a democracy like India, incidents of voter coercion and vote buying are some examples of such violations. [4] [5] [6]

1.2 Indian voting scenario

1.2.1 Evolution of voting process in India: India has gradually evolved in the way voting is carried out. The very earliest elections namely the General Elections held between October, 1951 and March 1952 were carried out by using paper ballots. These paper ballots were then replaced by the Electronic Voting Machines (EVMs). The EVMs were deployed on an experimental basis in 1982, their extensive use started in 1998 and in the 2004 General Elections EVMs were deployed across all polling stations thus putting an end to paper ballot. Nepal, Bhutan, Namibia and Kenya have purchased India-manufactured EVMs. Fiji was expected to use Indian EVMs in its elections in 2014. [7]. In 2013, however, the Supreme Court of India ordered that the

voter should be given the assurance that the vote casted by him/her has gone to the candidate of his/her choice.This led to the development of VVPAT (Voter Verifiable Paper Audit Trail) systems. The ECI in a discussion with journalists has hinted that in the 2019 Lok Sabha polls, 100% of the EVMs will be replaced by VVPAT systems.

1.2.2 Security Analysis of India's Electronic Voting Machines [8]: Hari K Prasad with his research team demonstrated how the current EVMs can be compromised [8]. They published a video showing how an attacker can replace the display of an EVM with a dishonest display that can be controlled remotely via Bluetooth. In another attack demonstration they overwrote the memory chip with their own data of the votes thus demonstrating a serious violation which can be exploited to turn the votes in any candidate's favour.

In their study they found out that there was no way to verify the integrity of the software running in the EVMs. They discovered that all votes stored in the EVM memory were stored with the date and timestamp with no encryption in place to protect this information. This was a serious violation of the voter's privacy. The paper also pointed out that the manufacturing of microcontrollers for the EVMs were carried out by private companies outside India. Hence, if the manufacturing process at that facility was compromised unknowingly then a lot of damage could be done to the entire voting process.

1.2.3 ECI Status paper [9]: In 2017 Election Commission of India released a descriptive document titled "Status Paper on Electronic Voting Machine (EVM)". The main aim was to defend again all the allegations raised by various parties and researchers. Though the paper attempted to cover all the aspects of the EVM but it still failed to address some of the following issues:

- The absence of a mechanism to verify that the code running on the machine is authentic and has not been tampered in any way.
- The security measure in place to protect the local storage where voting data is stored which can compromise voter secrecy.

Introduction of VVPAT also introduces a new vulnerable point in the system as the VVPAT system itself will have vulnerabilities in addition to the EVM's vulnerabilities. Another serious issue that needs consideration is that currently voter authentication is done manually, this leads to many issues related to voter identity mismanagement [10], Aadhar would serve as a possible solution to this debacle and is also the part of our proposed system.

2. PROPOSED SYSTEM

2.1 System overview

The following sections describe the architecture, design, and deployment perspective of the voting machine prototype. The prototype was designed and built as an illustration of the proof-of-concept that the blockchain technology could indeed be used to conduct elections. The prototype cannot be directly adopted as a full-fledged voting system. It would need several customizations and changes to be incorporated that are in line with the voting process. Only after the design and implementation details are subject to stringent review by the security and technology community will the blockchain based voting machine be fit to be adopted for actual elections.

It is advisable that the hardware design and the source code of the voting system be maintained in public repositories to enable inspection and review by anyone interested in strengthening the system.
Some properties of the blockchain data-structure seem to solve the challenges that any voting system has to address. Here is how using the blockchain addresses some of these challenges:

1. Immutability of votes: Many popular blockchain platforms use the Merkle tree (some implementation also use other variants of the Merkle tree) to verify the integrity of the data added to the blockchain. Even if a single bit of data is altered or tampered, it can be easily detected using a Merkle tree verification. This property of the blockchain to ensure that a vote once added to the blockchain cannot be altered or tampered helps in achieving immutability as well integrity verification of the votes.
2. Mathematical proof of recorded votes: Through the use of public key cryptography and hash or message digest values, mathematical proof of the origination of the data, and the exact time at which the data was added to the blockchain database can be provided. The blockchain database here simply refers to the voting related data stored in the blockchain.
3. Data redundancy: The blockchain is synced across all the nodes participating in the blockchain network, and hence this provides data redundancy. Even if data at multiple locations is corrupted, tampered or even destroyed it can be recovered from any node on the network.

Deploying the system on a protected private network would significantly reduce the risk of attacks on the system.

2.2 Architecture

Each node in the network is an individual voting unit. The term node and voting unit will be used interchangeably from here on. All nodes are a part of a private Ethereum network. Each node runs a geth node [11] that is a part of a private Ethereum network. Depending on the consensus protocol chosen, each node can also mine. Each node runs a local web server. The server is essential as it can communicate with the fingerprint module for biometric authentication of the voter as well as the geth node. The frontend GUI that is visible to the voter is built using HTML, CSS and Javascript (including the JQuery library). It must be noted that the server is not configured to serve the static files (HTML files, image resources, font files, etc.). In the prototype's implementation the browser loads the static files directly, and the requests to the server become cross domain requests [12]. This could be a security issue [13], but the server itself can be configured easily to serve the static files and remove the need for enabling cross domain requests on the browser. Please refer figure 1.

To summarize, the user interacts with the system using the HTML based GUI, these interactions trigger requests to the server, the server communicates with the geth node and the fingerprint module connected to the node as necessary. The Web3.py [14] python

module is used to request information from the geth node and send commands to the geth node. To enable such communication between a python program and the geth node, the required Ethereum APIs have to be enabled through command line arguments [15] when starting the geth node.

The prototype's implementation also includes a monitoring node, that can be used to monitor the entire network, provide an overview of the blockchain network, and also the number of transactions and votes recorded in the blockchain. The monitoring node can also be used to initiate the elections as well as terminate them.

The prototype uses a touchscreen as a medium of user interaction. A touchscreen based interaction model makes the system flexible and accessible. The touchscreen can be configured to display all the text in any regional language that the voter chooses.

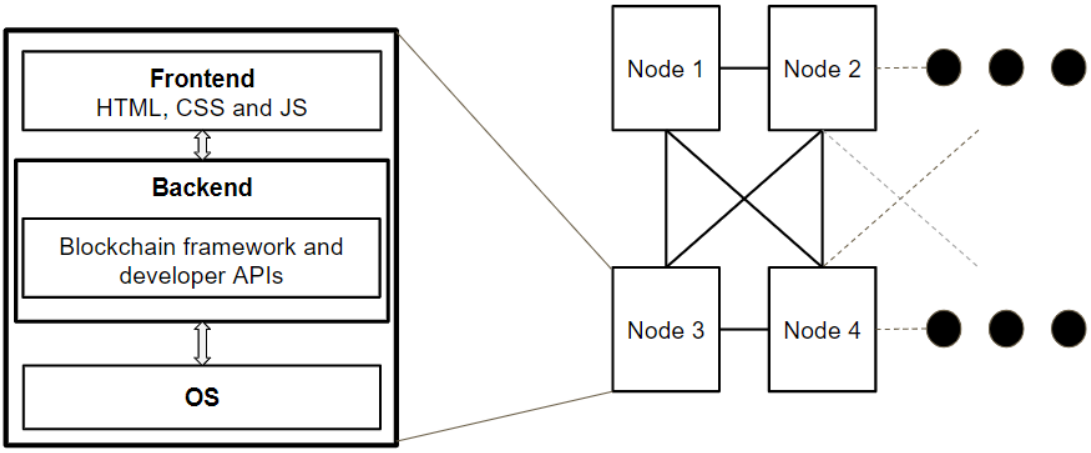

Fig. 1: System architecture

2.3 System operation

2.3.1 Setting up a private network: It is recommended that the actual system be deployed on a private network, and that the nodes be allowed to connect to the network only after a proper authentication process. The authentication credentials could be hard coded on individual devices or given to the officers responsible for setting up the individual machines on voting locations.

2.3.2 Uploading candidates lists: In the prototype's implementation the candidate lists are hard coded in the smart contract deployed on the network. This guarantees that the list cannot be altered once the contract is deployed. Thus, for each zone or ward (Election Commision of India terminology) [9], a separate smart contract has to be deployed on the network with candidate list for that specific ward. An alternative is to add all the lists of candidates for all wards in a single smart contract and then nodes in a ward would deal with the respective candidates for that ward.

2.3.3 Starting up individual nodes: First, the geth node is started with the required APIs enabled. Once the node is up and running, the local server is started. The server communicates with the geth node through the Web3.py module. The first job of the server is to connect the geth node to the private Ethereum network. Once the geth node is connected to other peers in the private Ethereum network [16] the device is ready to accept votes.

2.3.4 The voting process: Figure 2 depicts the whole voting process.

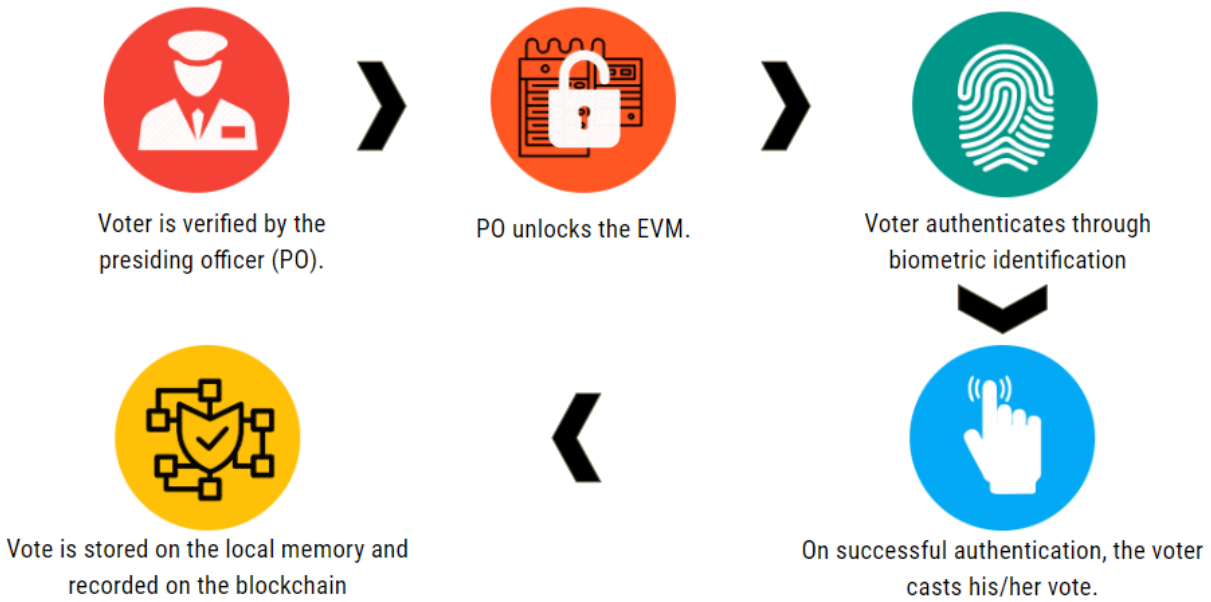

Fig. 2: Voting process flow

As shown in the figure above the steps in the voting process are:

1. The Presiding officer unlocks the voting machine. (Figure 3)

Fig. 3: Fingerprint verification of the presiding officer

2. Then the voter's identity is established using biometric authentication. (Figure 4)

Fig. 4: Fingerprint verification of the voter

3. The voter is then presented with a list of candidates. The voter chooses a candidate. (Figure 5)

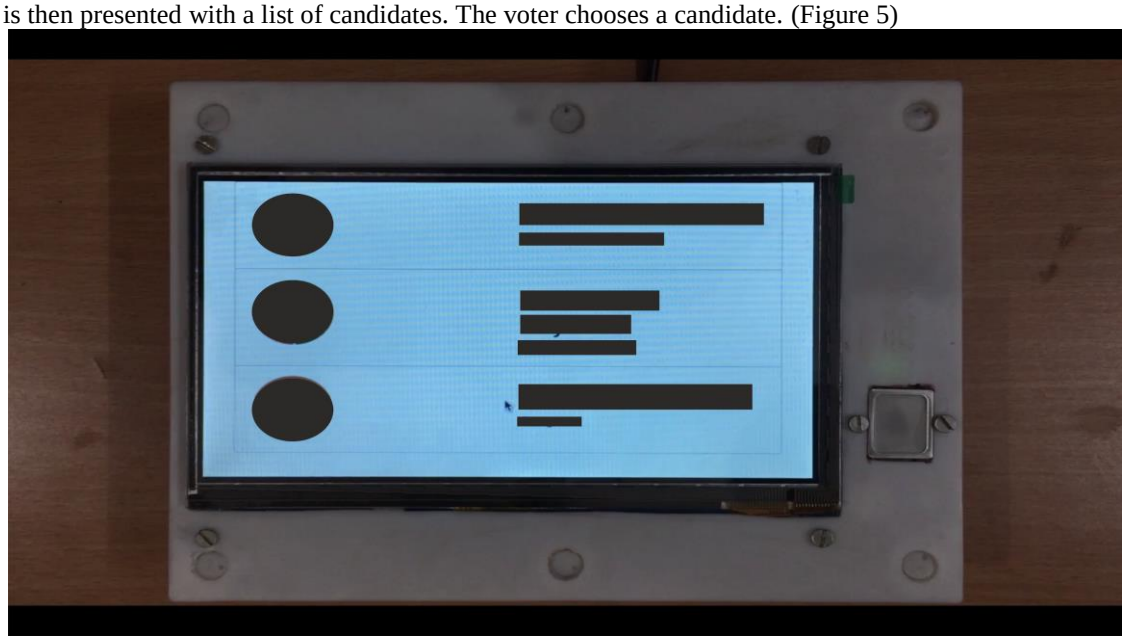

Fig. 5: Candidate list presented to the voter

4. The local web server ensures that the transaction that encapsulates the casted vote is added to the blockchain. The voter is given acknowledgement that his/her vote is successfully recorded. (Figure 6)

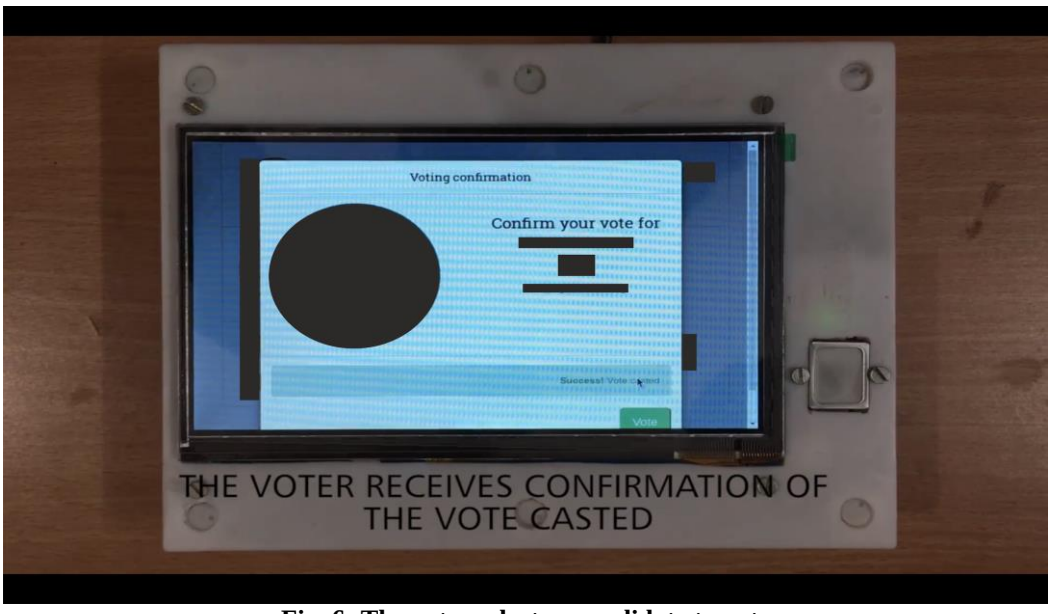

Fig. 6: The voter selects a candidate to vote

5. The voting machine is locked.
Once the presiding officer unlocks the machine the vote must be cast before a preset timeout, if the timeout expires, the machine is locked and the presiding officer must unlock it again to enable voting. "Locking of the machine" refers to locking the underlying Ethereum account. An unlocked Ethereum account is essential to sign and broadcast the Ethereum blockchain transactions [17]. Casting of a vote is a simple increment operation performed on the number of votes associated with the candidate that voter voted for, no other information is captured.

Randomly generated token values are used to establish a voting session to prevent any forgery attempts.

2.4 Drawbacks, challenges and possible solutions

2.4.1 Ensuring voter anonymity: Even though the system does not record any data that can identify a voter and the vote he/she casted, it might be possible to link a voter to his/her vote. This is due to the fact that each transaction that carries a vote (basically a simple increment operation) also has a timestamp [17], and it is also possible to identify the Ethereum account that made the transaction. So, if a single Ethereum account is associated with a single node a vote can be traced back to a specific voting unit, and with the timestamp it is possible to guess the voter that cast the vote.

To overcome this drawback, a bank of different Ethereum accounts could be used. A single bank will be shared by a specific number of voting units. Each voting unit, for casting a vote, will randomly pick an account from this shared bank and use that account to make a transaction (adding the vote to the blockchain). This removes the possibility of tracing back a transaction to a singular voting unit. However, an issue with this is that if a re-election has to be carried out at a particular location. Then re-elections will also have to be carried out at all the locations where the voting machines that share the account bank with that single machine are located.

2.4.2 Biometric database storage: Another issue is the location of storage of the biometric database. Considering the Indian scenario, if the Aadhar APIs are used for biometric authentication of voters and the voting units have to communicate with the Aadhar server via the Internet then such a link from the private protected network of voting units and the Internet could be a potential entry point for intruders. It is therefore desirable to store the database on a node in the private protected network of the voting units, of course, with adequate security measures to prevent unauthorized access to the data on the database.

2.4.3 Publicly known vulnerabilities in open source software: Most of the software used in the current implementation is open source software. Use of open source software is a double edged sword. Open source software is easily accessible to anyone who wants to review the code. This generally results in better quality code and software. But, a major drawback of using open source software in a high stake scenario like elections is the public knowledge of discovered vulnerabilities. Any security vulnerability that is discovered in an open source software are generally discussed on forums or broadcasted on mailing lists, if such a discovery is made a few days before an election or even on the Election Day, then upgrading and patching the vulnerability overnight would be difficult. Since the security vulnerability is public knowledge, attempts could be made to gain access to the system by exploiting the vulnerability.

2.4.4 Codebase maintenance: This issue could be seen as an extension to the previous point. If open source software are adapted to make voting systems then the election conducting authority must and cannot rely solely on the open source community to maintain the software and issue updates. Enough funds and resources must be allocated for periodic review and updation of the software stack used in the voting system.

2.4.5 Establishing the private network: Establishing private networks for conducting elections would require little or no investment in setting up the infrastructure as private networks could be established using the existing pervasive cellular infrastructure in India. Technologies like CDMA could provide required connectivity in remote regions.

2.4.6 Alleviating some security concerns: The current implementation uses interpreted JavaScript scripts and Python scripts. There are some security concerns that arise with the use of technologies like JavaScript [18]. To improve security in this aspect one alternative is to build the application on the voting unit as a native application and create a single binary executable. The permission on the filesystem could be configured in such a way so as to allow execution of only the executable through the use of file permission flags. No other file in the non-root user space will be allowed to execute.

Message digest or hash values could be used to verify if the code running on individual voting units is the code deployed by the election conducting authority and that malicious code has not been injected.

This could be achieved by calculating the hash values of the files before they are deployed on the voting units. After the voting units are deployed at their respective location with the code/script files on them, the election conducting authority could remotely calculate the hash values of the files on each of the deployed voting units and cross-check it with pre-calculated hash values. This process of ensuring integrity of deployed code becomes more convenient if a single executable is used instead of a large number of script files.

If the private network of voting units is established properly the threat of network attacks like man in the middle attacks, masqueraders and DoS or DDoS attacks can be thwarted. Depending on the reports of a security review of the private network infrastructure intrusion detection and prevention system (IDS) could also be deployed.

3. CONCLUSION

The implemented system addresses most of the issues with the legacy system as explained below:

1. Due to the use of a distributed database the concept of point of failure does not apply to this system. The legacy system had multiple critical points of failure.
2. All the data is stored using encryption algorithms: The Ethereum framework used in the implemented system uses the Elliptic Curve Cryptography for encryption purposes. The broadcasted transactions are signed using ECDSA [19]. The nodes use their public key to sign the transactions. The private key is used for encryption. The legacy system stored votes in plaintext.
3. The system is failure resilient as data is available on all the nodes in the network, hence data corruption even at multiple points won't lead to loss of data. The data is present on every node as the blockchain is basically a distributed database.
4. Authentication of the presiding officer and the voter is completely digital and based on biometric signature, hence bogus voting is not possible. Due to reliance on manual authentication there was a possibility of bogus voting in the legacy system.
5. The process of calculating results and verifying them is completely automatic and any risk of errors involved in the manual process are avoided. The legacy system relied on a manual process for counting of votes and evaluation of election results, which was prone to human errors.
6. The proposed system is cheaper than the currently used EVMs. The legacy EVMs cost about 10,000 INR per EVM [20]. But, the newly implemented system prototype costs around 8000 INR.
7. Votes once recorded in the blockchain are immutable. As there is an implicit verification of the Merkle tree on every replication of the blockchain, any unauthorized attempt to forge or insert data can be easily identified and resolved.

A credible voting system enforces the strength of the democracy. Our implemented and tested system attempts to address various issues plaguing the existing system and reinforce the faith of the voters in the democratic process, through the use of digital systems and mechanisms.

In conclusion, the implemented system is a secure and robust voting system based on blockchain.

4. ACKNOWLEDGMENT

We would like to take this opportunity to thank our guide Prof. Dr. J. S. Umale from PCCOE, Pune, Mr. Hari Duche from Persistent Systems Ltd. and the Department of Computer Engineering, PCCOE for giving us all the help and guidance we needed. Their keen interest inspired us to complete our presentation in a successful manner. They offered us all the independence we needed and at the same time ensured we were on the right track.

We wish to express our gratitude to all the people who directly or indirectly guided us towards the successful completion of our project. We also thank the open source community that actively and tirelessly works to develop and maintain the various frameworks, operating systems, programming languages and libraries used in this project. The work of that community solely driven by passion is what made this project a reality.

The voting system described in this text is the subject of Indian patent application no. 201821015256.

5. REFERENCES

[1] Monobina Gupta, 2018, How India Became Democratic. Retrieved from https://thewire.in/books/india-became-democratic
[2] Election Commission of India. The Function (Electoral System). Retrieved from https://www.eci.nic.in/eci_main1/the_function.aspx
[3] National Democratic Institute, 2013, Secrecy. Retrieved from https://www.ndi.org/e-voting-guide/secrecy

[4] Election Commission of India. Handbook for candidates. Retrieved from https://www.eci.nic.in/archive/handbook/candidates/cch7/cch7_1.htm
[5] Rahul Verma, 2017, Death of patronage? Retrieved from https://www.thehindu.com/opinion/op-ed/death-of-patronage/article18525772.ece
[6] Rahul Verma and Harsh Shah, 2017, Money can't always buy votes. Retrieved from https://www.thehindu.com/opinion/op-ed/money-cant-always-buy-votes/article20461124.ece
[7] Wikipedia community. Electronic voting in India. Retrieved from https://en.wikipedia.org/wiki/Electronic_voting_in_India
[8] Hari K. Prasad, J. Alex Halderman, Rop Gonggrijp, Scott Wolchok, Eric Wustrow, Arun Kankipati, Sai Krishna Sakhamuri, and Vasavya Yagati. April 29, 2010. *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010.* Retrieved from https://indiaevm.org/evm_tr2010.pdf
[9] Election Commision of India. Status Paper on Electronic Voting Machine (EVM). Retrieved from https://www.eci.nic.in/eci_main1/current/StatusPaperonEVM_09052017.pdf
[10] Praveen Chakravarty, April 2014, The mysteries of 'dead' and 'deleted' voters. Business Standard. Retrieved from https://www.business-standard.com/article/opinion/praveen-chakravarty-the-mysteries-of-dead-and-deleted-voters-114042700690_1.html
[11] The Ethereum Community. Geth documentation. Retrieved from https://github.com/ethereum/go-ethereum/wiki/geth
[12] Mozilla Developer Network. Web Docs. Cross-Origin Resource Sharing (CORS). Retrieved from https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
[13] Jordi Giménez, Which Security Risks Do CORS Imply? Retrieved from https://mobilejazz.com/blog/which-security-risks-do-cors-imply/
[14] Piper Merriam, Jason Carver, Web3.py documentation. Retrieved from https://web3py.readthedocs.io/en/stable/
[15] The Ethereum Community. Command Line Options in go-ethereum. Retrieved from https://github.com/ethereum/go-ethereum/wiki/Command-Line-Options
[16] The Ethereum Community. Private Network in go-ethereum. Retrieved from https://github.com/ethereum/go-ethereum/wiki/Private-network
[17] Mahesh Murthy, December 2017, Life Cycle of an Ethereum Transaction. Retrieved from https://medium.com/blockchannel/life-cycle-of-an-ethereum-transaction-e5c66bae0f6e
[18] Liam Tung, March 2017. An insecure mess: How flawed JavaScript is turning web into a hacker's playground. Retrieved from https://www.zdnet.com/article/an-insecure-mess-how-flawed-javascript-is-turning-web-into-a-hackers-playground/
[19] Hartwig Mayer, ECDSA Security in Bitcoin and Ethereum: a Research Survey. Retrieved from https://pdfs.semanticscholar.org/1785/6bad4335c8ca7419aab2c715ea25ce5e0621.pdf
[20] Press Trust of India, March 2014, Electronics Corp, Bharat Electronics get EVM contracts. Retrieved from https://indianexpress.com/article/business/companies/electronics-corp-bharat-electronics-get-evm-contracts/

