International Journal for Modern Trends in Science and Technology, 7(01): 73-76, 2021

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST070116
Available online at: http://www.ijmtst.com/vol7issue01.html

Online Voting System - Powered by Blockchain

Vaibhav Sinha[1] | Sachin Garg[2]

[1]UG student, IT & Maharaja Agrasen Institute Of Technology,Delhi,India
[2]Asst. Prof. Department of IT & Maharaja Agrasen Institute Of Technology,Delhi,India

To Cite this Article
Vaibhav Sinha and Sachin Garg, "Online Voting System - Powered by Blockchain", *International Journal for Modern Trends in Science and Technology*, Vol. 07, Issue 01, January 2021, pp.- 73-76.

Article Info
Received on 22-November-2020, Revised on 18-December-2020, Accepted on 28-December-2020, Published on 03-January-2021.

ABSTRACT

Electronic voting or e-voting has been used in varying forms since the 1970s with fundamental benefits over paper based systems such as increased efficiency and reduced errors. However, there remain challenges to achieve widespread adoption of such systems especially with respect to improving their resilience against potential faults.

Blockchain is a disruptive technology of the current era and promises to improve the overall resilience of e-voting systems. This paper presents an effort to leverage benefits of blockchain such as cryptographic foundations and transparency to achieve an effective scheme for e voting. The proposed scheme conforms to the fundamental requirements for e-voting schemes and achieves end-to-end verifiability. The paper presents details of the proposed e-voting scheme along with its implementation using the Multichain platform. The paper 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, Blockchain, Cryptographic Foundations,Transparency*

INTRODUCTION

Elections are a 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. 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. 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.

Overview

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 blockchain technology to seek solutions to these challenges.

METHODOLOGY

In this study, I propose the use of blockchain technology as a database used to securely store casted votes while preventing any form of tampering and multi-factor authentication to verify the eligibility while granting the voters permission/access to cast their votes. A framework would also be proposed which would enable eligible voters to easily and securely cast their votes using their mobile devices from any location while also increasing voters' trust towards the voting process and offer better transparency during the voting process.

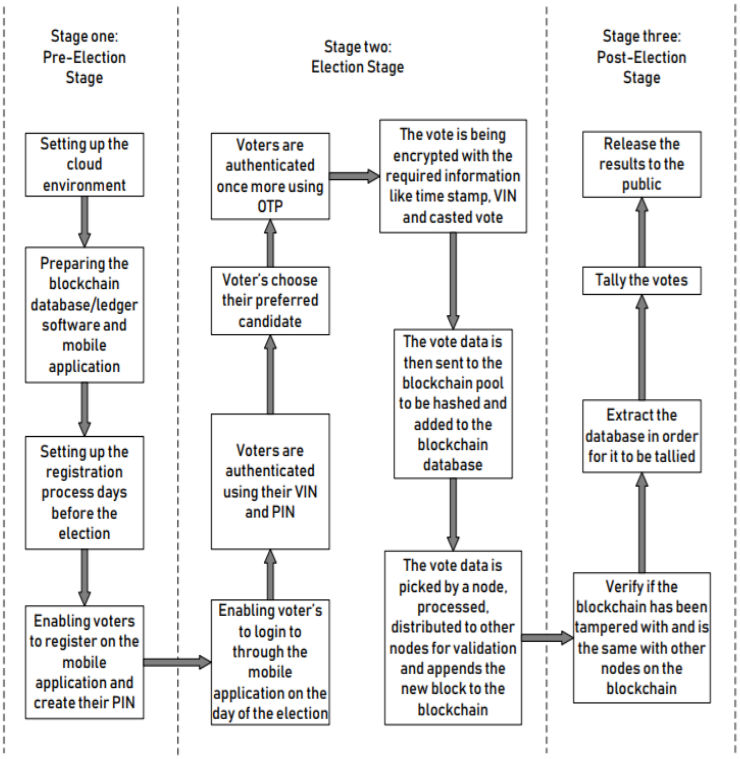
The existing framework made use of a centralized server that houses the database which makes it a simple point of failure that could bring down the whole system. The model utilized the three phases from the adapted framework and utilized a database that held every voter's identification number (VIN) alongside their phone number and other important data. The proposed system made use of a distributed blockchain database instead of a single centralized database and also made use of multi-factor authentication to verify the voters which include their voter's identification number (VIN), PIN and one-time password (OTP). The proposed framework is presented in Figure.

1. Front End Layer: This layer is categorized into two (2) phases which are the pre election phase (can be utilized by the voters to register themselves and is stored in the voter's database) and the election phase (voters using their mobile devices can log onto the website and can cast their votes). The Front-End will comprise the **Registration Page , Login Page, Candidate Details Page, UserDetails Page, etc**. The Technologies used would be ReactJs, Redux,

BootStrap, Html, CSS, JavaScript.

2. Back End Layer: This layer deals with the blockchain database that can be viewed and monitored by the EMBs which is carried out at the post-election phase. Here, the casted votes are stored in the blockchain distributed database and once the election phase is over, the votes would be tallied and the final result would also be presented or shown. This proposed framework would provide a tool for increasing the overall efficiency of the electoral process, advancing democracy, adding credibility to election results, and building trust in electoral management. It would achieve all of this using a three stage process which is explained in Figure below.

The technologies used will be Nodejs, ExpressJs, DataBase, Web3JS, Smart Contracts, Solidity, Truffle Framework, Ethereum, etc.



Index of functionalities

Below, we will elaborate the functionalities of a novel ballot and election smart contract for an e-voting system, without the integration of a government identity verification service.

```
contract CreationElection {

address[] public deployedBallots;

constructor (bytes32[] candidates,
bytes32[] district, uint hours) public {
for(uint i = 0; i < district.length; i++){
address newBallot = new
Ballot(candidates, district[i],
```

```
msg.sender,                    hours);
deployedBallots.push(newBallot);
 }
}

function getDeployedBallots() public view
returns(address[]) {
return deployedBallots;
} }
```

• **ElectionCreation constructor**: Takes in a list of candidates and districts along with the address of the wallet of the creator and the amount of hours the election will take. The constructor then creates a single smart contract for each district provided and puts the address of each smart contract created into the deployedBallots array.

• **getDeployedBallots**: returns an array with the address of each created smart contract in each index of the array.

```
contract Ballot {

struct Voter {
 uint weight; // weight is accumulated by
delegation
 bool voted;   // if true, that person
already voted
address delegate; // person delegated to
uint vote;  // index of the voted candidate
}
struct Candidates {
bytes32 name;
uint voteCount;

uint creationDate;
uint expirationDate;
 }

Candidates[] public candidates;

 address public chairperson;

 mapping(address => Voter) public voters;

constructor(bytes32[]              memory
candidateNames) {
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

for (uint i = 0; i < candidateNames.length;
i++) {
```

```
 candidates.push(Candidates({      name:
candidateNames[i],     voteCount:     0,
creationDate: now, expirationDate: now +
amountOfHours
 }));
 }
    }
```

• **Ballot constructor**: Sets the manager of the ballot smart contract to the address of the wallet which created the election, the voting district of the smart contract to the district which the ElectionCreation contract provided and then proceeds to fill the Candidates struct with the list of candidates provided and the number of votes for each candidate to 0. The constructor also stores the time of the creation of the contract along with the time when the contract is to expire.

```
modifier restricted() {
 require(msg.sender == chairperson);
 _;
}
```

• **Restricted modifier:** This modifier is used to restrict functions in the manner that only the creator of the election can access the information which the functions give.

```
function vote(uint candidateId) public {
 Voter      storage      sender      =
voters[msg.sender];

require(sender.weight != 0, "Has no right
to vote");

 require(!sender.voted,     "Already
voted.");
 sender.voted = true;
 sender.vote = candidateId;

candidates[candidateId].voteCount   +=
sender.weight; }
```

Vote: This function allows voters to vote. The requirement for a voter to vote is that the voted attribute of the voter should be set to default false.If that is the case, the function guarantees that the election time limit has not been reached. If both requirements are satisfied, the contract retrieves the index of which candidate was voted for and increases his vote count by 1 and sets the

voted attribute to true, so that the voter can never vote again in this particular election.

SECURITY ANALYSIS

1) DDoS: To successfully DDos a distributed system such as we have proposed, the attacker must DDoS every single bootnode in the private network. The individual or institution would be immediately located if that would occur. Each node is implemented with a Byzantine fault tolerance algorithm, which helps locating failed nodes in the system.

2) Authentication vulnerability: Each individual is identified and authenticated by the system by presenting an electronic ID from Auðkenni and the corresponding 6-digit PIN in the voting booth. Without supervision, an individual could vote for multiple people, if the individual had knowledge of the PIN for each corresponding electronic ID he has. To further address this vulnerability in the near future, a biometric scan could be introduced.

3) Sybil: Sybil attack is known against centralized systems, where an individual creates a large amount of nodes in an attempt to disrupt network operation by hijacking or dropping messages. Since our proposal is running in a private network no individual has the access to create one. Even the consensus protocol that is used in our system is prone to Sybil attacks. Private blockchains solve many of today's security problems using strong cryptography features and

the limited access to the ledger, without negating the transparency aspect the blockchain technology offers.

CONCLUSION & FUTUREWORK

Electronic voting has been used in varying forms since the 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 the 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 in- depth evaluation of the approach highlights its effectiveness with respect to achieving fundamental requirements for an e-voting scheme. In continuation of this work, we are focused at improving the resistance of blockchain technology to the '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 achieved which motivates us to investigate it further. 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] G. Schryen, "Security Aspects of Internet Voting," in IEEE, Hawaii, 2004.

[2] Wikipedia, "List of controversial elections," 20 September 2016. [Online]. Available: https://en.wikipedia.org/wiki/List_of_controversial_elections. [Accessed 27 September 2016].

[3] N. Uribe, "10 Benefits of Electronic Voting," 01 August 2016. [Online]. Available: http://www.fobsoftware.com/blog/10-benefits-of-electronic-voting-for-home-ownerassociations. [Accessed 28 September 2016].

[4] 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.

[5] 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.

[6] Vitalik Buterin. (2015). Ethereum White Paper. Available at: https:// github.com/ethereum/wiki/wiki/White-Paper

[7] Ayo, C. K., Ekong, U. O., Ikhu-omoregbe, N. A., & Ekong, V. E. (2007). M-voting implementation: The issues and trends. 1–5. Retrieved from http://www.academia.edu/download/3258019/EEE4041.pdf