# Electronic voting System using Blockchain

ELEN E6883

Topics in Signal Processing: Introduction to Blockchain

Spring-2022

Name: Uday Mukhija

UNI: um2158

## Abstract

Online voting as a phenomenon is gaining momentum as well as importance in modern society. It has great potential to decrease organizational costs, increase transparency and increase voter turnout. It eliminates the need to print ballot papers or open polling stations—voters can vote from wherever there is an Internet connection. Despite these benefits, online voting solutions are viewed with a great deal of caution because they introduce new threats.

The first aspect of our design is the registration process, verifying a voter is essential in establishing security within the system. Making sure that someone's identity isn't being misused for fraudulent purposes is important, especially when voting is considered, where every vote matters. A single vulnerability can lead to large-scale manipulations of votes. Electronic voting systems must be legitimate, accurate, safe, and convenient when used for elections. Nonetheless, adoption may be limited by potential problems associated with electronic voting systems. Blockchain technology came into the ground to overcome these issues and offers decentralized nodes for electronic voting and is used to produce electronic voting systems mainly because of their end-to-end verification advantages. This technology is a beautiful replacement for traditional electronic voting solutions with distributed, non-repudiation, and security protection characteristics. For a sustainable blockchain-based electronic voting system, the security of remote participation must be viable, and for scalability, transaction speed must be addressed.

## Introduction to Decentralized Application:

Blockchain technology provides a platform for creating a highly secure, decentralized, anonymized yet auditable chain of record. In this project, this has been used to record and report votes and prevent voter fraud in elections.

The project has been made using the following applications:

1. Remix IDE

2. Ganache
3. Metamask Wallet

The Blockchain Structure is also known as an append-only data structure, such that new blocks of data can be written to it, but cannot be altered or deleted. Private blockchain limits the read and write access, only specific participants can verify their transactions internally. That makes the transaction on a private network cheaper, since they only need to be verified by a few nodes that are trusted and with guaranteed high processing power. Nodes are very well-connected and faults can quickly be fixed by manual intervention, allowing the use of consensus algorithms which offer finality after much shorter block times. In this project, I have used Permissioned Blockchain which will use the Proof of Authority(PoA) consensus Algorithm. A Consensus Algorithm is used to set restrictions on selected known entities to certify and validate transactions on Blockchain. Here, this will help us to stop adding new people without Administrators Permission. This Algorithm Proves to be helpful because it does not leak the Voter's Information and Voting Data.

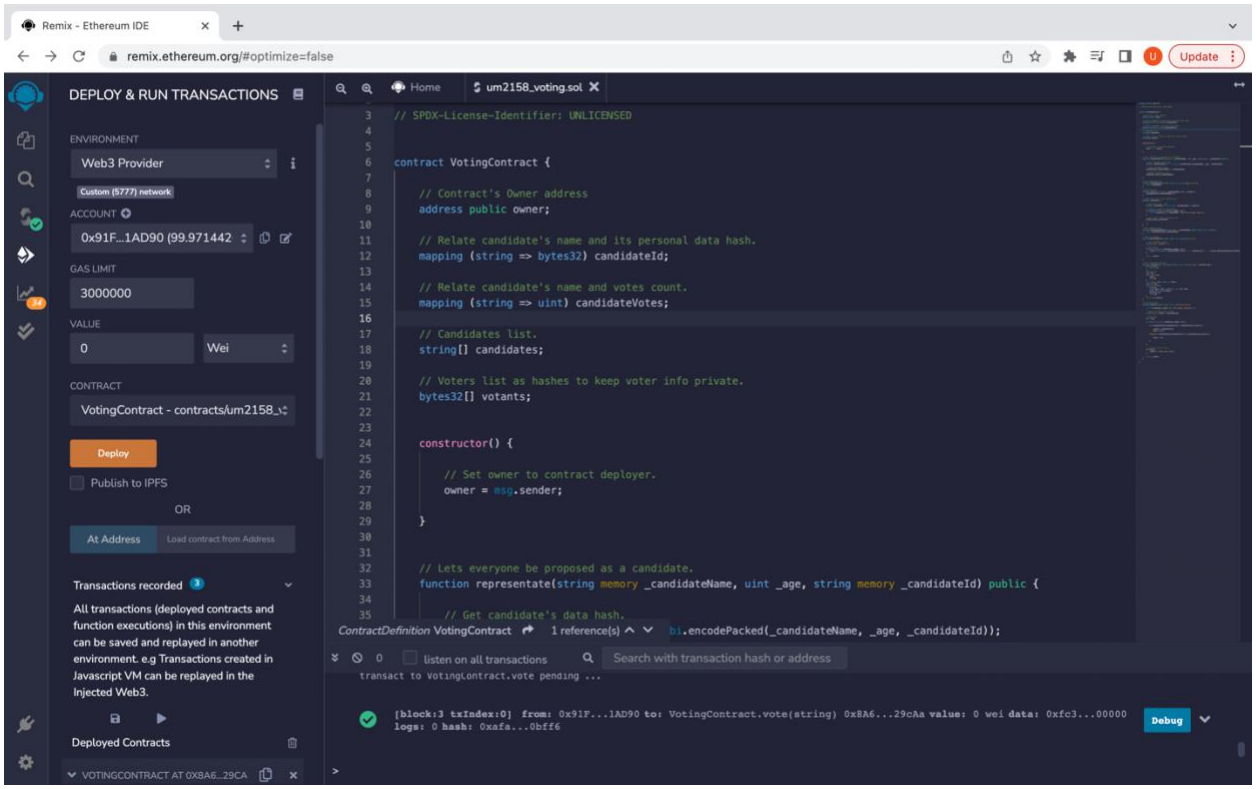
The smart contract will be linked to the wallet (in the sample run shown below, it is linked with a Metamask wallet). The user types in the candidate ID, age and name. Then they proceed to voting for the candidate of their choice. Each user can only vote once. The user can check the candidate they voted for, as well see get a breakdown of votes in the voting results, see owners of the votes cast and see the winners of the election.

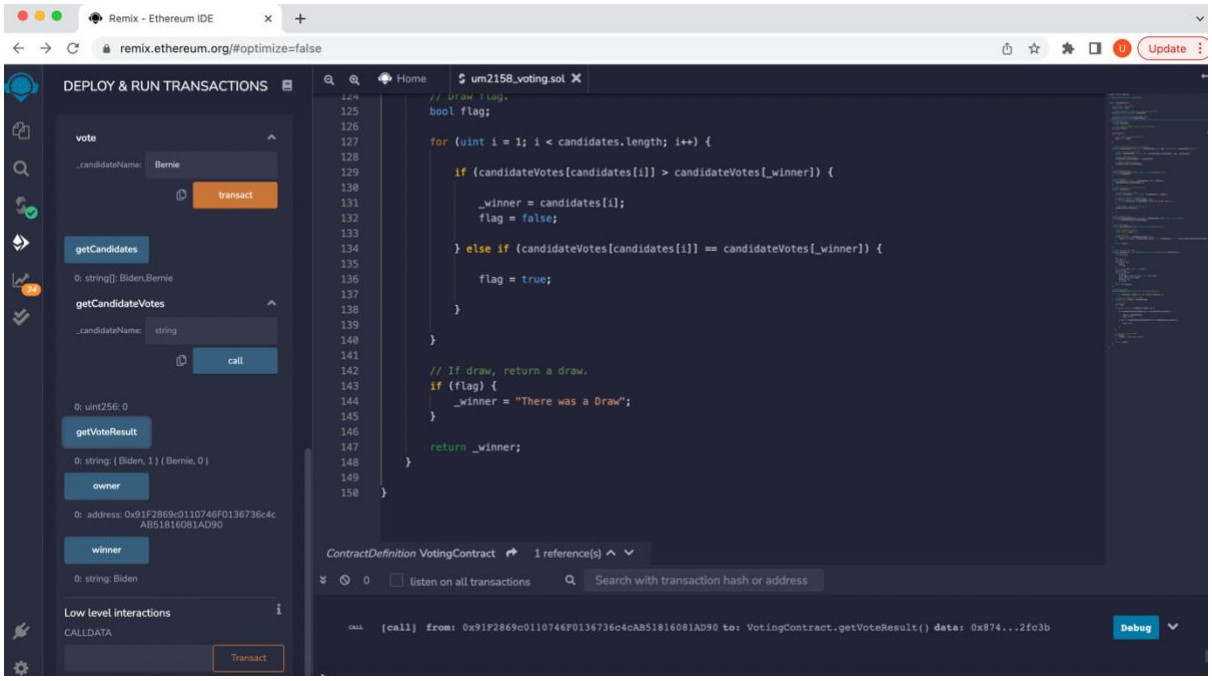
## Originality:

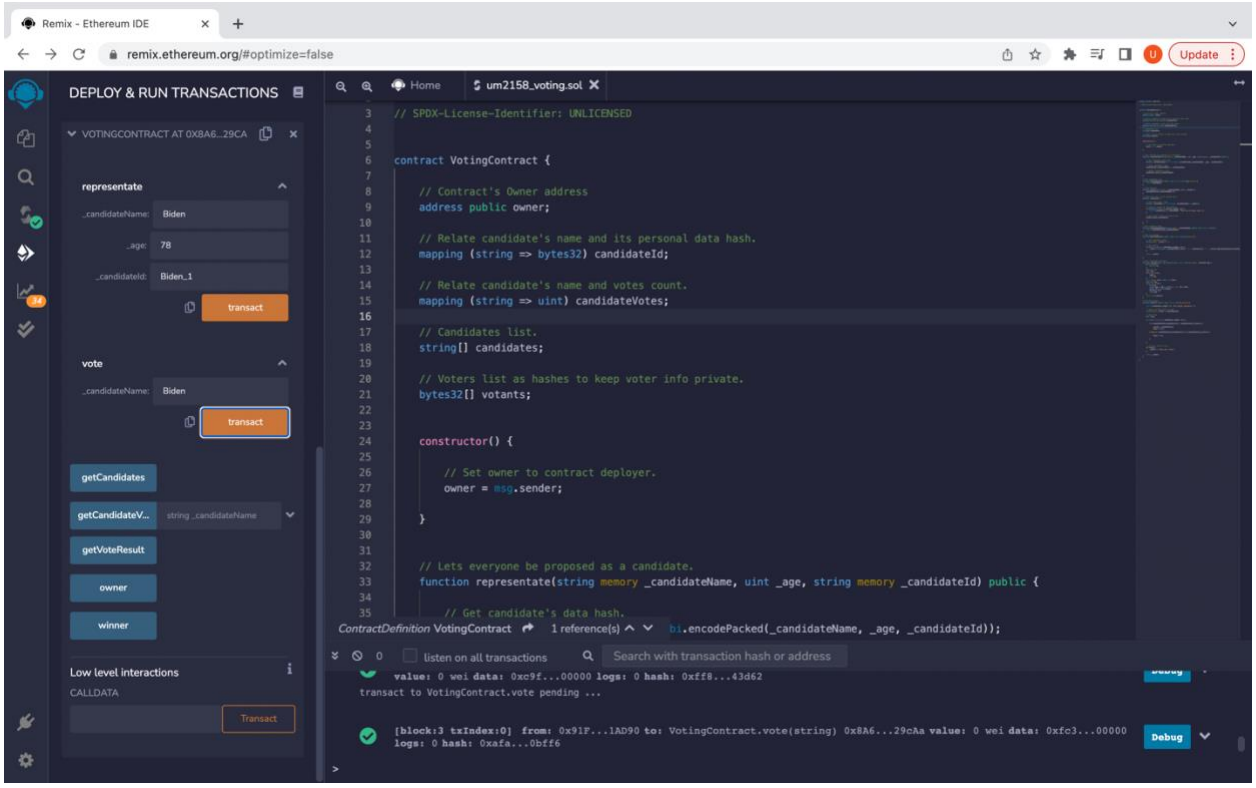
Currently, in most parts of the world, voting either takes place through a ballot or through Electronic Voting Machines (EVMs). These EVMs are of two of types, namely remote voting system and a direct recording system. The direct recording system architecture consists of a power unit, control unit, display unit and a voting unit. However, there are constant rumours and conspiracy theories about the EVM machines, as well as ballot votes, if one is to look at how political discourse shapes in large democracies like India and the United States of America.
Compared to existing EVM machines, the smart contract in the project offers fast synchronization, secure channel and enrollment control.
Also, in the literature review, it was found that present smart contracts worked with Exonum, which uses rust programming language and isn't user developer friendly. So I used Solidity. The vote can be viewed publicly, it is easily verifiable along with the person

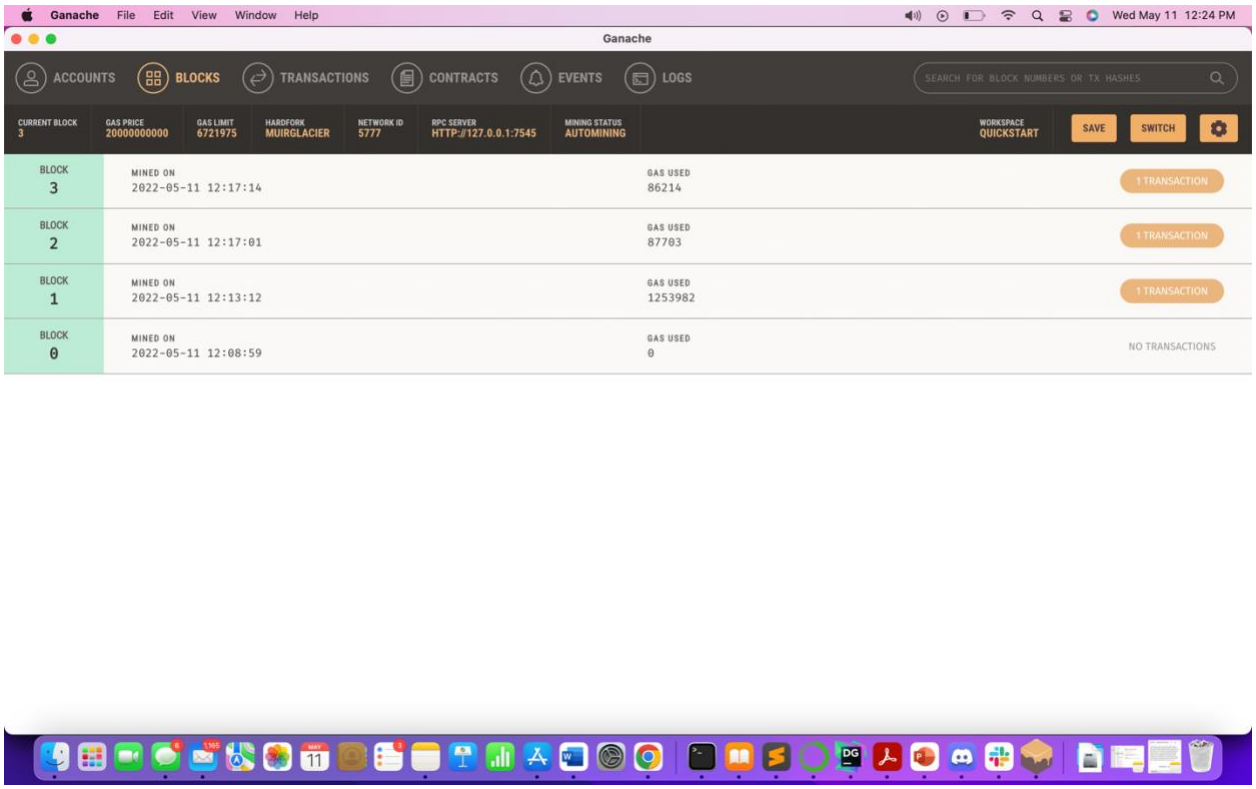
making it and it cannot be changed once it is given a unique hash. Apart from voter verification, a distributed system like this the attackers will have to DDoS every single boot node in the private network, which can be immediately located. No individual also has the access to create a large number of nodes for a Sybil attack in the system proposed in this project.

## Sample Run:




Remix - Ethereum IDE
remix.ethereum.org/#optimize=false
Update
DEPLOY & RUN TRANSACTIONS
vote
_candidateName: Bernie
transact
getCandidates
0: string[]: Biden,Bernie
getCandidateVotes
_candidateName: string
call
0: uint256: 0
getVoteResult
0: string: ( Biden, 1 ) ( Bernie, 0 )
owner
0: address: 0x91F2869c0110746F0136736c4cAB51816081AD90
winner
0: string: Biden
Low level interactions
CALLDATA
Transact
Home
um2158_voting.sol
// Draw flag.
bool flag;
for (uint i = 1; i < candidates.length; i++) {
if (candidateVotes[candidates[i]] > candidateVotes[_winner]) {
_winner = candidates[i];
flag = false;
} else if (candidateVotes[candidates[i]] == candidateVotes[_winner]) {
flag = true;
}
}
// If draw, return a draw.
if (flag) {
_winner = "There was a Draw";
}
return _winner;
}
}
ContractDefinition VotingContract
1 reference(s)
listen on all transactions
Search with transaction hash or address
[call] from: 0x91F2869c0110746F0136736c4cAB51816081AD90 to: VotingContract.getVoteResult() data: 0x874...2fc3b
Debug


Remix - Ethereum IDE
remix.ethereum.org/#optimize=false
Update
DEPLOY & RUN TRANSACTIONS
VOTINGCONTRACT AT 0X8A6...29CA
representate
_candidateName: Biden
_age: 78
_candidateId: Biden_1
transact
vote
_candidateName: Biden
transact
getCandidates
getCandidateV...
string _candidateName
getVoteResult
owner
winner
Low level interactions
CALLDATA
Transact
Home
um2158_voting.sol
// SPDX-License-Identifier: UNLICENSED
contract VotingContract {
// Contract's Owner address
address public owner;
// Relate candidate's name and its personal data hash.
mapping (string => bytes32) candidateId;
// Relate candidate's name and votes count.
mapping (string => uint) candidateVotes;
// Candidates list.
string[] candidates;
// Voters list as hashes to keep voter info private.
bytes32[] votants;
constructor() {
// Set owner to contract deployer.
owner = msg.sender;
}
// Lets everyone be proposed as a candidate.
function representate(string memory _candidateName, uint _age, string memory _candidateId) public {
// Get candidate's data hash.
bi.encodePacked(_candidateName, _age, _candidateId));
ContractDefinition VotingContract
1 reference(s)
listen on all transactions
Search with transaction hash or address
value: 0 wei data: 0xc9f...00000 logs: 0 hash: 0xff8...43d62
transact to VotingContract.vote pending ...
[block:3 txIndex:0] from: 0x91F...1AD90 to: VotingContract.vote(string) 0x8A6...29cAa value: 0 wei data: 0xfc3...00000
logs: 0 hash: 0xafa...0bff6
Debug




Conclusion:

We have created a decentralized voting system that promises increased transparency, eliminates fraud and rigging, shows results in real time and empowers voters and shareholders. It makes remote voting easier and reliable, thereby leading to a more direct and inclusive democracy.

GitHub Link:

https://github.com/um2158/evoting-using-blockchain/tree/main