

# Cryptography (BITS F463)

## Blockchain for Electronic Voting System



Dasoju Pranay Kumar

2019A7PS0006H

Venkata Sai Ram Prabhat Mutyala

2019A4PS1331H

Raja Vardhan Reddy Siriganagari

2019A7PS1337H

## **Abstract:**

Electoral integrity is essential not just for democratic nations but also for state voter's trust and liability. The traditional or paper-based polling method served to increase people's confidence in the selection by majority voting. Engineers across the globe have created new voting techniques that offer some anti corruption protection while still ensuring that the voting process should be correct. . Electronic voting increases election reliability when compared to manual polling. In contrast to the conventional voting method, it has enhanced both the efficiency and the integrity of the process. Because of its flexibility, simplicity of use, and cheap cost compared to general elections, electronic voting is widely utilized in various decisions.

Despite this, existing electronic voting methods run the danger of over-authority and manipulated details, limiting fundamental fairness, privacy, secrecy, anonymity, and transparency in the voting process. Most procedures are now centralized, licensed by the critical authority, controlled, measured, and monitored in an electronic voting system, which is a problem for a transparent voting process in and of itself.

Our project aims to solve some of the problems with electronic voting using blockchain technology.

## **Problem Statement:**

Our aim is to solve some of the problems of electronic voting using blockchain and zero knowledge proofs. Blockchain ensures anonymity of voter and decentralization whereas zero knowledge proofs ensure anonymity of candidates to whom votes have been cast and fairness of the election process.

## Blockchain:

Blockchain is a peer-to-peer network that build by a chain of blocks. A cryptographic hash and timestamp are added to the preceding block in blockchain. Cryptography is a secure networking approach that uses computer science and mathematics to hide data and information from others, allowing data to be transmitted securely across the insecure network, in encrypted and decrypted forms

## Approach:

Blockchain technology corrected flaws in today's election methods, making the polling process clear and accessible, preventing unlawful voting, strengthening data security, and verifying the polling results.

One does not have central ability. The data is stored on various nodes; there is no central authority. It's impossible to hack all of the nodes and make changes to the data. As a result, it is impossible to destroy the votes, while also effectively verifying them by tallying votes with other nodes.

Several conditions must be met, whether we're talking about traditional paper-based voting, voting via digital voting machines, or voting via an internet voting system.

**Eligibility** - To vote online, voters must use a recognised identification system to verify their identity. All legitimate voter's identifiers must be added to the list of participants. However, there are dangers: double-checked to ensure that no illegal voters are added, and the identification system should be both trusted and secure.

**Unreusability** - After a voter casts their ballot, all that is required is to make a mark in the participation list and deny them the opportunity to vote again.

However, privacy must be considered; consequently, guaranteeing both unreusability and voter anonymity is difficult.

**Privacy** - When it comes to online voting, privacy means that no one except the voter knows how a voter voted.

**Fairness** - Fairness in terms of no one getting intermediate results is simple to achieve: Before sending, voters encrypt their choices, which are then decrypted at the end of the voting process. The important point to understand here is that if someone has access to encrypted judgments and a decryption key, they can get intermediate results.

**Soundness and Completeness** - We must demonstrate that the encrypted data satisfies the requirements of a legitimate ballot without disclosing any information that could aid in determining how the vote was cast. Zero-knowledge proof is used to tackle this problem.

### **Zero Knowledge Proof:**

- Zero knowledge proofs are used in our project mainly to ensure the anonymity of candidates to whom the vote is being cast.
- The vote given by each voter is encrypted using a hashing algorithm.
- A value of 1 is put against a candidate if the voter did vote for the candidate and 0 otherwise.
- These values are then summed up at the end of the election to find out the results of the election.
- We need to implement two main features to ensure the fairness of these elections.
- Firstly, the sum of values given to all the candidates by each voter must be 1.
- Secondly, the value given to a particular candidate must be binary, either 0 or 1.

- We use the following zero knowledge proof and run it for 5 rounds to verify that the vote is a valid vote.

1. The voter chooses a random number  $0 \leq r < p - 1$  and sends it to the miner as  $h = g^r \text{mod}(p)$ .
2. The miner receives  $h$  and sends back a random bit  $b$  (could be 0/1).
3. The voter sends  $s = (r + bx) \text{mod}(p - 1)$  to Bob.
4. The computes  $g^s \text{mod}(p)$  which should equal  $hy^b \text{mod}(p)$ .

If the zero knowledge proof doesn't fail, the miner is sufficiently convinced that the voter is voting for the first time and that the vote is a valid vote.

## Working of each party in the election:

### Voter

- **castVote()**:- Each voter logs in and enters their user id and casts their vote using this function.
- **viewUser()**:- View the details of the user. This function displays the voter id and the encrypted choice of the candidate.

### Miner

- **processTransactions()**:- This function processes each vote and adds it to a new block. If the block contains the maximum number of transactions(votes) it can hold, the mineBlock() function starts the mining process.
- **mineBlock()**:- This function starts the mining process and adds a new block to the blockchain if the mining process is successful.

### Trusted third party

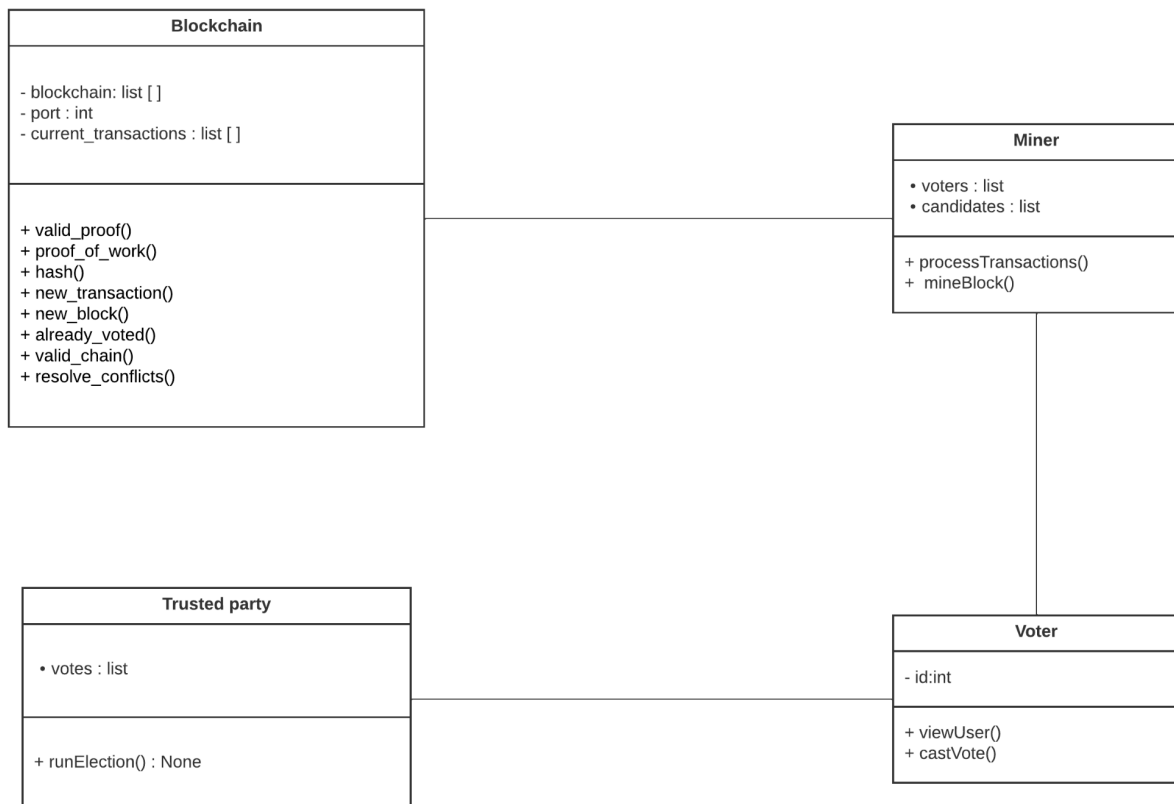
- **runElection()**:- Collect votes from each of the voters and declare the election results.

## Blockchain class:

This class contains all of the functionality of the blockchain.

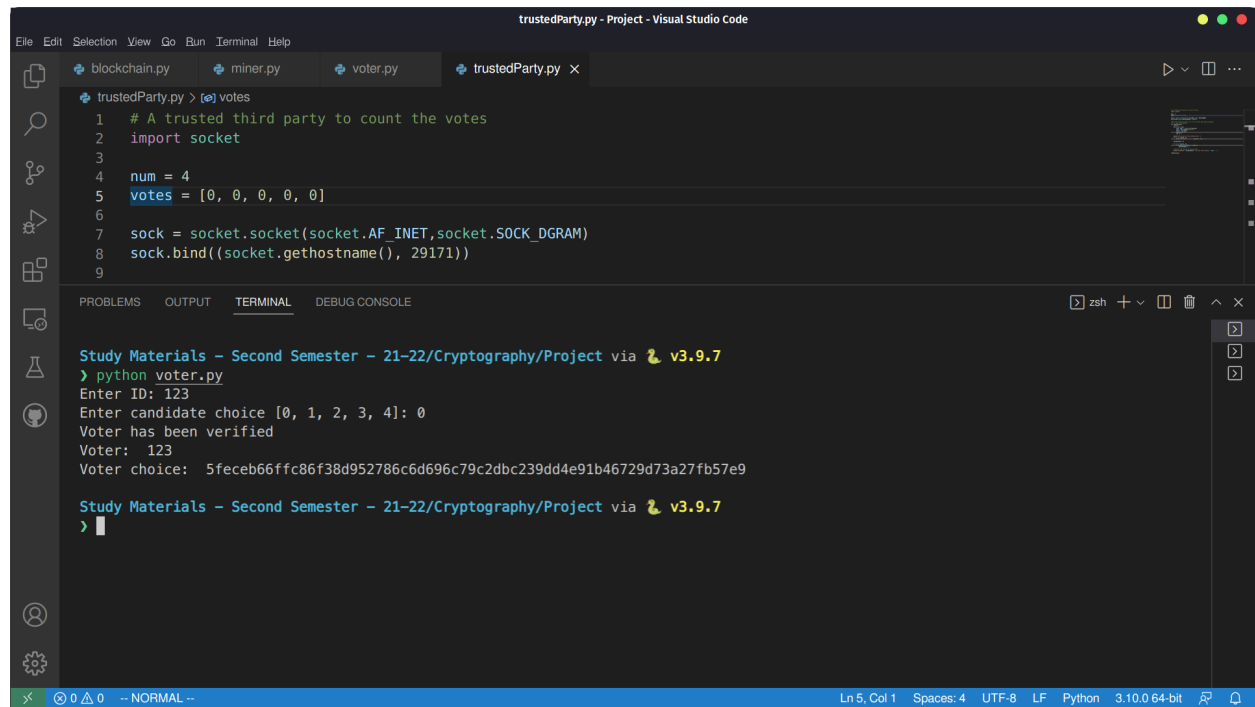
- **resolve\_conflicts()**:- Pick the maximum length chain as the valid chain.
- **valid\_chain()**:- Verify that the chain is valid.
- **already\_voted()**:- Make sure that the voter is voting for the first time by traversing the blockchain.
- **new\_block()**:- Create a new block to be added to the blockchain
- **new\_transaction()**:- Add the transaction to the latest unverified block.
- **hash()**:- Hash the block given as argument.
- **proof\_of\_work()**:- Main proof of work algorithm
- **valid\_proof()**:- Check if the trial nonce validates the proof.

## UML Diagram:



## Working application:-

Voter:



The screenshot shows a Visual Studio Code editor window titled "trustedParty.py - Project - Visual Studio Code". The editor has four tabs: "blockchain.py", "miner.py", "voter.py", and "trustedParty.py". The "trustedParty.py" tab is active, displaying the following Python code:

```
1 # A trusted third party to count the votes
2 import socket
3
4 num = 4
5 votes = [0, 0, 0, 0, 0]
6
7 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
8 sock.bind((socket.gethostname(), 29171))
9
```

Below the editor, the "TERMINAL" panel is open, showing the output of a command prompt session. The prompt is "Study Materials - Second Semester - 21-22/Cryptography/Project via v3.9.7". The user has entered "python voter.py". The output shows the program running and accepting input:

```
> python voter.py
Enter ID: 123
Enter candidate choice [0, 1, 2, 3, 4]: 0
Voter has been verified
Voter: 123
Voter choice: 5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9

Study Materials - Second Semester - 21-22/Cryptography/Project via v3.9.7
>
```

The status bar at the bottom indicates the file is at "Ln 5, Col 1", uses "Spaces: 4", "UTF-8" encoding, "LF" line endings, and is a "Python 3.10.0 64-bit" file.

Miner:

The screenshot shows the Visual Studio Code editor with the file `trustedParty.py` open. The code defines a trusted third party to count votes using a socket. The terminal output shows the program's execution, including the initialization of rounds, voter verification, and the final election results.

```
trustedParty.py > votes
1 # A trusted third party to count the votes
2 import socket
3
4 num = 4
5 votes = [0, 0, 0, 0, 0]
6
7 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
8 sock.bind((socket.gethostname(), 29171))
9
```

Terminal Output:

```
Round 2 initiated...
Round 3 initiated...
Round 4 initiated...
Round 5 initiated...
Voter has been verified
Round 1 initiated...
Round 2 initiated...
Round 3 initiated...
Round 4 initiated...
Round 5 initiated...
Voter has been verified
Hash found: 00009edba7f518a560dac40db4808a4105fcfd9f2ec97ca25de6ed17fcf2c9a8
Added New Block
Index: 2
Timestamp: 1650910405.5750427
Transactions: [{'voter': '5423', 'voted_for': 'd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35'}, {'voter': '43', 'voted_for': '4b22777d4dd1fc61c6f884f48641d02b4d121d3fd328cb08b5531fcacdaf8a'}]
Proof: 12641
```

Trusted party:

The screenshot shows the Visual Studio Code editor with the file `trustedParty.py` open. The code is the same as in the previous screenshot. The terminal output shows the program's execution, including the initialization of rounds, voter verification, and the final election results.

```
trustedParty.py > votes
1 # A trusted third party to count the votes
2 import socket
3
4 num = 4
5 votes = [0, 0, 0, 0, 0]
6
7 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
8 sock.bind((socket.gethostname(), 29171))
9
```

Terminal Output:

```
Study Materials - Second Semester - 21-22/Cryptography/Project via v3.9.7
> python trustedParty.py
The results of the election are:
Candidate 0 : 1
Candidate 1 : 0
Candidate 2 : 2
Candidate 3 : 0
Candidate 4 : 1
Candidate 2 has won the election!
Study Materials - Second Semester - 21-22/Cryptography/Project via v3.9.7 took 46s
>
```