Protecting democracy with a trustless blockchain based decentralised election system

Sumuk Shashidhar

March 11, 2021

Abstract

Democracy fades as sophisticated attempts of voterfraud are detected, with some even succeeding. VoteBlock attempts to protect democracy by decentralising the election process to ensure the lack of a single point of failure or control, with the help of a blockchain. It must be understood that while VoteBlock secures the election process, it does not secure the voter registration process essential for authorizing each voter.

Contents

1 Introduction

2 Structure of A Block

2.1 Components

Each block consists of the following elements essential to recording votes and identifying induviduality of each voter.

1. Index - To enumerate each block
2. Transactions - The given transactions in each block (votes)
3. Timestamp - The timestamp of each block's creation
4. Previous Hash - The hash of the previous block to facilitate the blockchain
5. Nonce - The number only used once. Used to supplement the rest of the data to generate a desired hash pattern

2.2 Hashing

SHA-256 hashing is used, where the data of the block is first dumped into a string, and then computed after being unicode encoded.

2.3 Code Used

block.py

```
from hashlib import sha256
import json

class Block:
        def __init__(self, index, transactions, timestamp, previous_hash, nonce=0):
                        self.index = index
                        self.transactions = transactions
                        self.timestamp = timestamp
                        self.previous_hash = previous_hash
                        self.nonce = nonce

        def compute_hash(self):
                block_string = json.dumps(self.__dict__, sort_keys=True)
                return sha256(block_string.encode()).hexdigest()
```