Protecting democracy with a trustless blockchain based decentralised election system

Sumuk Shashidhar

Janurary 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 Code Used

block.py

```
from hashlib import sha256
import json

class Block:
        def __init__(self, index, transactions, timestamp, previous_hash, nonce=0):
                        """This function initialises the block of the blockchain
                            using the regular concept

                        Args:
                        index ([type]): Index number of the block
                        transactions ([type]): The Transactions to be stored in the
                             given block
                        timestamp ([type]): The timestamp of the given block
                        previous_hash ([type]): The hash of the previous block to
                            store
                        nonce (int, optional): The number only used once. The
                            nonce is to maintain uniqueness, making it hard to
                            regenerate, which gives the blockchain the power it
                            needs. Defaults to 0.
                        """
                        self.index = index
                        self.transactions = transactions
                        self.timestamp = timestamp
                        self.previous_hash = previous_hash
                        self.nonce = nonce

        def compute_hash(self):
                """
                A function that return the hash of the block contents.
                """
                block_string = json.dumps(self.__dict__, sort_keys=True)
                return sha256(block_string.encode()).hexdigest()
```