Conducting Trustworthy, User Friendly Elections in Large Democracies

Using public permissioned blockchain

Team Name : Consensus

Team Details :

1. Kunal Jain
2. Prince Varshney
3. P Sahithi Reddy

Background

India is one of the largest democratic nations in the world and general elections play an important role in democracy. Currently Elecetion Commission, an independent constitutional body of India conducts these elections. It is largely centralized and there are various concerns like if all the votes are correctly counted and revealed etc. Hence there is need of some degree of transparency in the system, without violating the anonymity of user. Blockchain based solutions offer interesting ways to consduct elections.

Properities Followed

We indetify the following properities as important for conducting general elections.

1. **Secret Ballot (SB)**: No one should be able to trace a vote on the blockchain back to the voter who cast it. This is an essential condition in representative democracies to ensure there is no voter intimidation or bribery.
2. **Sealed Result (SR)**: The results of the election should be sealed until voting is over. This ensures that the people voting first in the elections are not at any disadvantage than those voting afterwards.
3. **Consistent Results (CR)**: There should be verifiability that no votes are being miscounted.
4. **1 Person 1 Vote (1P1V)**: No one should be able to vote twice. This is a feature of equality present in most democracies.
5. **Authorized Voting (AV)**: No foreign or non-participatory entity should be able to cast a vote. This ensures that the elections of a region or country are not influenced by outside support or interests.
6. **Unchangeable Votes(UV)**: No voter should be able to change their votes after they have been added to the blockchain.
7. **Non Transferable Votes (NTV)**: Voters should not be able to transfer their votes to other voters.
8. **Publicly Verifiable (PV)**: After conducting the elections, everyone in the public should be able to verify that the election results are correct. This means everyone should be able to verify all the other conditions on their own

Hidden Coin Approach

In this approach there are two types of coins Dummycoins and Votecoin.

Registering :

Election Commission (EC) first starts accepting candidates (`addCandidates`) and changes it state to accepting candidates and then voters (`add voters`) by changing its state to acceptvoters, thats equivalent to EC registering people. With each change in election state, it is transmitted on blockchain.

Functions :

1. **acceptCandidates** - EC changes the state
2. **addCandidate** - adds candidates to the election
3. **acceptVoters** - EC changes state to accepting voters
4. **addVoter** - adds voters to election

Coin Allocation

EC allots some amount of dummycoins to everyone and there would be one votecoin. It is realised by generating random number and choosing that as votecoin and it is communicated to EC and voter. Number of dummycoins is atleast as many as contesting parties.

Functions :

1. **setDummyCoinValue** - to set how many dummy coins
2. **getVoteCoinIndex** - gets to know what index is vote coin at

Voting

Voters send the coin to candidates after election state is `RUNNING`. The send voteoin for who they truly want to vote and dummy coins to others, where the coins arent distinguishable.

Functions :

1. **sendCoin** -sends coins from voter to candidates
2. **startElection** - EC can start the election

Reveal & Counting

Voters reveal their votecoin index, EC cross checks and counts the votes.

Functions :

1. **getVoteCoinIndex** - gives vote index index for a voter.
2. **startReveal** - EC starts reveal state
3. **endElection** - EC ends election
4. **getWinner** - Counts number of votes to give winner among contested candidates

Hidden Party Approach

In this approach each party has multiple accounts, each voter gets one account for each party and voters will have different combinations.

Registering :

Election Commission (EC) first starts accepting candidates (`addCandidates`) and changes it state to accepting candidates and then voters (`add voters`) by changing its state to acceptvoters, thats equivalent to EC registering people. With each change in election state, it is transmitted on blockchain.

Functions :

1. **acceptCandidates** - EC changes the state
2. **addCandidate** - adds candidates to the election
3. **acceptVoters** - EC changes state to accepting voters
4. **addVoter** - adds voters to election

Account Allocation

EC allocates to voters one account corresponding to each voter to which they can send their vote to.

Functions :

1. **setNumberofAccounts** Number of accounts for each party/candidate.
2. **startElection** - EC starts the election
3. **getCandidateAccount** - for a voter and candidate combination.

Voting

Votes vote after election is started, to accounts they were allocated to.

Functions :

1. **sendVote** -sends the votecoin to the candidate he wants to vote to.

Results

EC releases list of which account belongs to who, and runs count to get winner.

Functions :

1. **startReveal** - EC starts reveal state
2. **endElection** - EC ends election
3. **getWinner** - Counts number of votes to give winner among contested candidates aggregating all accounts.