AnonEvote
A Project on Blockchain Evoting
Functional Specification

Michael Wall

Saturday 19th November, 2016

Version	**Comments**
0.1	Initial Draft.

1 Introduction

This document describes an implementation of an electronic voting system which utilises a blockchain[1] database. The system is used to allow a user to electronically cast a vote in an anonymous, verifiable and tamper-proof manner. Votes are ecnrypted and cast over a peer-to-peer network. The final votes can be counted and verified by any user, but no users' votes can be traced back to a voter.

The need for this system arises from the lack of a secure, trustless, tamper-proof and anonymous electronic voting system. Paper ballots are slower, and more expensive to conduct (work hours to organise polling stations, tallying votes, re-counts of votes, multiple voting options, errors filling out a ballot, etc). The AnonEvotesystem aims to tackle some of these issues.

Blockchain The blockchain is the database which maintains the *ledger* of *votes*. It is distributed amongst all participants in the system via peer-to-peer networking. Each client will maintain an up-to-date version of the *ledger* to the best of their knowledge. The blockchain is described further in section §3.2 on page 5.

Ledger The *ledger* is a growing record of each *vote* which has been cast. It is publicly viewable.

Vote Each block in the *ledger* contains a single *vote*, which is encrypted to provide anonymity and to prevent tampering. The structure of a secure *vote* is described further in §3.1 on page 4.

2 General Description

2.1 Product/System Functions

The AnonEvotesystem is designed to enable electronic voting to be carried out securely, tamper-proof and anonymously. A user of the system will cast a vote electronically through some user interface. The user's vote is then encrypted to form a *secured vote*. The user can verify that their vote is what they intended by decrypting it. This will spoil the ballot, but after repeated casting and decrypting the user can be sure that their vote is accurate. When the user is happy with their ballot, it is then broadcast to all of the client's peers to be verified. The verification of the transaction requires a proof of work to be a valid block. This proof of work entails some computationally expensive task. When the proof of work is complete, the block is then broadcast to a client's peers. If there is disagreement between a new block, consensus is used to select the correct version which is then broadcast to all peers.

To verify their vote, a user can look at the blockchain and verify that their receipt exists on the chain. Homomorphic encryption of votes allows them to be tallied without decrypting individual votes. Any user will be able to perform the computation to tally the votes.

2.2 User Characteristics

A user will be any eligible voter. The users are not expected to require any prior knowledge of blockchains, cryptography or other technical concepts described in this document. Because different users may have different user needs or certain levels of ability, the system must be accessible. As a graphical user interface is not a major priority for the system, this challenge will not be addressed to any major extent. The system will be used by text inputs, similar to a command line program, on top of which a UI can be built at a later date.

A user should be able to input a vote, and verify their selection. Once the user has made their selections, they should be able to validate the encryption of their vote as many times as they see sufficient. Once the user is happy that the system is registering their intended vote, the user should then be able to cast their vote to the network to have the transaction verified and added to the ledger. The user should also receive a digital receipt of their transaction so that they can verify their vote at a later date. The user should also be able to perform the required computation to tally the votes should they wish to do so.

2.3 Operational Scenarios

2.3.1 Main

The main scenario in which the system is intended to be used is any large scale elections, referendums or other votes in which a large body of people will participate in. These votes are required to provide anonymity to voters, tamper-proof security, easy tallying and re-counting of ballots, and a trustless[2] environment.

2.4 User stories

2.4.1 User casts a vote which is valid

A user selects candidate A on their electronic ballot. They verify that their selection was correctly registered and cast the vote. Their vote is then verified by peers on the network and added to the ledger.

2.4.2 User spoils their ballot by selecting wrong options

The user selects both candidate A and B on their electronic ballot. The system will not allow an incorrect selection such as this, and alerts the user to the error. The user then starts the process from the start.

2.5 Constraints

Different voting systems[3] have different requirements.

In the simplest case a voting system in which a voter selects a single option from two or more choices, and the option with the most votes wins. Other systems are more complicated and require more complicated computation to achieve a result.

Such systems include weighted voting systems, proportional, semi-proportional, rated and multiple winner systems. Such systems pose challenges as the votes may not be very easily calculated as with a simple tally.

Each voting system would require its own rules to specified for the vote to be successful. For this reason the AnonEvotesystem will focus on two systems as a proof of concept, with the ability to add more systems. The two systems will be *Proportional Reprisentation with a Single Transferable Vote*[4] and *Party List Proportional Representation*[5].

3 Functional Requirements

3.1 Secure Votes

Here is format of a secure vote.

3.2 Blockchain

Here is info on the blockchain

3.3 Requirement 1

Here is some other requirement

3.3.1 Description

3.3.2 Criticality

3.3.3 Technical issues

3.3.4 Dependecies on other requirements

3.3.5 Others

4 System Architecture

5 High Level Design

6 Preliminary Schedule

7 Appendices

Notes

[1] A blockchain is a distributed database which maintains a growing ledger of records called blocks. Each block is timestamped and linked to the previous block to create the chain. The data in a block cannot be altered without breaking the chain

[2] A trustless environment is one in which no one individual or group requires to be trusted to ensure the integrity of the system. For example, a bank is not trustless, as you must trust the bank to keep its record of your account accurate and free from tampering.

[3] A voting system consists of the set of rules which must be followed for a vote to be valid, and defines how otes are cast, counted and aggregated to yield the final result. Examples include a plurality, majority representation and many other variations.

[4] The PR-STV system is used in Ireland.

[5] The Party List PR system is used in 85 countries