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, recounts 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 3.

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 3.

2 General Description

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.

2.1 Product/System Functions

2.2 User Characteristics

2.3 Operational Scenarios

2.4 Constraints

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

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