Chester Taylor
Kartik Sathappan
Michael Whelton

Principles of Computer Security Group Project
Design Document

Project goal: To design and implement a secure electoral system, specifically one based on the distributed public ledger concept used in bitcoin and similar technologies.

1. What has been done
   - 1.1. Initial research into bitcoin's public ledger (the "blockchain"): how it works, how it is implemented.
   - 1.2. Initial research into potential software tools which could be of use in our implementation. One such tool is BigchainDB, which combines big data (useful in the context of keeping track of millions of voters) with the decentralization and immutability of the blockchain (the foundation of our security model).

2. What will be done
   - 2.1. A distributed public ledger which will hold information about votes (the "votechain")
   - 2.2. A client through which voters will cast their vote, and may view votes already in the votechain.
   - 2.3. A means by which some governmental body may distribute votes to voters (ensuring that each voter may vote exactly once).

3. How it will be done
   - 3.1. Votechain
     - 3.1.1. To save on work and time, the votechain will be implemented through an existing open-source tool. The top candidate (currently under investigation) is BigchainDB.
   - 3.2. Voter client

3.2.1. Voters will interact with the votechain through a web client. Login will be secured through two-factor authentication (the user must supply a code which will either be texted or emailed to him, or he may use an app such as Google Authenticator).

3.2.2. The web client will present voters with a means of selecting which candidate they wish to vote for.

3.2.3. Once a user casts his or her vote, the client will encrypt the vote data, open a communication channel to the network, and broadcast the encrypted packet. A replay attack of one of the vote packets would be analogous to a double-spend, which BigchainDB protects against; therefore, the votechain will be replay-attack resistant.

3.2.4. Voters must be able to look their vote up on the votechain; to this end, the web client will display a "vote transaction ID" when a user casts his vote. The votechain should support querying transaction IDs for information about those transactions (voter ID, candidate ID). For ease-of-use, the client should also have functionality to automatically query the votechain for the user, so that the user does not have to manually look up his voter ID/transaction ID. More discussion on voter IDs will be found in section 3.3.

3.3. Government administration

3.3.1. Whichever authoritative body oversees the election must be able to grant an equal number of votes (we shall assume exactly one vote) to each voter at the beginning of each election. Voters will receive their votes as part of the "genesis block" in the votechain: the initial transaction will assign each voter his allotment of votes.

3.3.2. Voters will have a unique "voter ID," which will identify their vote on the votechain. The voter ID is roughly analogous to a spending address in bitcoin, as votes are sent from a voter ID to a candidate ID.

3.3.3. In order to ensure the one-to-one correspondence between voters and their IDs, voter IDs will be derived from a public-private key pair (similar to in bitcoin), where the private keys are derived using a salted hash of the voter's Social Security Number (SSN) as a seed value. Malicious parties should not be able to reverse the derivation process to obtain voters'

SSNs, but voters should be able to derive their own ID, and the administrative body should be able to derive voter IDs.