Rashidat Agbaje
Yichao Song
JaTara Brown
600.443
March 6, 2003

Project Design

Components

The overall electronic voting system that we designed can be broken down into the follow components: Voter Registration and Voting on Election Day

Voter Registration

The voter registration system that we designed requires the establishment of a government-funded organization called the Department of Voter Registration (DVR). The DVR will be responsible for registering voters and the creation and maintenance of the Registered Voters Database (RVB). The RVBs at each DVR will have a database(s) consisting of all the registered voters in the state. The databases at the DVR will communicate with each other using Spread, a toolkit for distributed systems that allows for secure group communication. (We will discuss spread in greater detail in a little bit.) It will also be the responsibility of the DVR to maintain the databases; it should check eligibility of the voters and voters should be able to check if their information is correct. This can be implemented by having the DVR send confirmation of the voter's registration (either electronically or by snail mail), which the voter can verify the information. In the event, that the voter finds incorrect data, the voter can either go to the DVR in person or by mail; the DVR can verify by signature.

The databases themselves will contain the name, address, optional email address, the county each voter belongs to, and their voter ID number, that will be randomly

assigned. They will only contain statewide information; we do not deal with the voter going out of state.

Voting on Election Day

On Election Day, the voter must first go to the verification station manned by a poll site volunteer. The volunteer will have a laptop that can access the first database at the local DVR. In order for the voter to progress to the voting station, he must verify who he is by first signing the Register Voter Signature Book, and producing photo identification. Once the voter's identity has been established, he will be given his voter ID and sent to the voting station, which uses the Sensus protocol. If the voter cannot properly identify himself, he may be given a provisional ballot, which requires that each county have paper ballots for their own county and all the other counties. At the voter agent, featuring a GUI interface, the voter will enter his voter id number, and the voter agent will generate a public/private key pair for that voter. Then the voter agent will ask the voter which county does he belong to and will then provide the correct ballot of the voter. Once the voter has voted and the voter agent has verified that the voter is happy with his choices, the voter agent sends the sealed ballot with the voter's public key to the validator, which can verify, by accessing the RVB, that the voter has received the correct ballot. The validator will update the database by adding the voter's public key. Once the validator has accessed the voter's record, it puts a hold on it in such a way that no other database can access or alter that voter's record until it has verified that the voter's vote was counted. Once validation is complete, the sealed ballot is returned to the voter agent, which then works to get the vote, stripped of the voter's identity, to the tallier to count. Additional functionality will be added to the voter agent so that the voter agent will tell

the validator to release whatever record that it has locked, indicating the that voter's vote has been counted. Functionality will be added to the tallier, which in the case that the voter is not a local voter, will, in specified interval time periods, send the voter's vote to his home county, while keeping copy separate from the local votes. Therefore, the talliers in each home county will have to be able to communicate with each other.

Protocols

The two protocols that we use are Spread and Sensus. As we mentioned above, Spread is a tool for group communication with distributed systems. It was designed to be modular in two ways: support for multiple link layer protocols and support for multiple client interfaces. Spread provides ordering and reliability guarantees. The level of service for these guarantees can be adjusted accordingly. To use Spread, each member of the group that wants to communicate must run a daemon on their host machine, and through SAFE messages, they can establish membership and communicate. There is a risk that an adversary could trick Spread to making him a member of a group that he should not belong to. In spite this risk, Spread is still extremely useful and secure. It provides the following:

- Efficient and reliable message bus.
- Reliable multicast from any number of sender to lots of receivers
- Membership services that inform each component of an application of which other components are running, and enables easy recovery when some fail.
- Agreed ordering of messages to the group – all receivers receive messages sent to the group in exactly the same order.
- The technology enables on connection per process to send to all other 599 processes.
- Replicated databases are kept synchronized in such a way that a client can query or update any of them and the results will be the same as if only one copy existed.

- N-way fail-over tool for clusters of server. Ensures that there will always be a server to handle requests that arrive on any of the IP addresses that are publicly known for the cluster.
- Powerful, but simple API. Only six basic calls are required to utilize Spread.
- Enables the system to grow seamlessly without architectural changes.
- Allows unicast, multicast, and multiple query functions.
- Open source form; has been tested and evenly commercially developed.

Sensus protocol: separate machines, like embedded computer systems for each component of the Sensus protocol. So our design will have a voter agent machine, a tallier machine, and a validator machine, that will communicate in an Intranet; however, the validator at some point on the protocol will have to access the RVB, through a secure channel, most likely using SSL or Blowfish.

We decided that in communication between RVBs, data will be encrypted over a secure channel. We realize that this may be overkill, but with something as important as elections, this may be necessary to instill voter confidence in the electoral process. In the event that there is network failure, there will be dedicated phone lines to maintain communication between databases. In addition, each vote and the voter's record with verification that the voter voted with be time stamped so as to any possible duplicate votes while the network is down. (This assumes that all the machines are synchronized.) The votes that are counted when the network goes down will be stored in a separate place.

The audit/paper trail will be produced by the tallier. No receipt will be presented to the voter – the voter only gets an on-screen confirmation. The validator will update the RVB once the voter's vote has been counted.

Algorithm

Voter Agent:

- retrieve ballot for presentation to voter

- voter marks choices and confirms all votes cast
- voter encrypts ballot with private key
- voter blinds ballot
- voter transmits ballot to validator for verification

Validator:

- accepts encrypted ballot and voterID from voter
- accesses DVR record corresponding to voter ID
- marks the validated bit to 1 to indicate validation occurred
- signs the encrypted ballot and transmits back to voter agent

Voter Agent

- transmits signed, encrypted, blinded ballot to tallier

Tallier

- accepts signed, encrypted, blinded ballot from voter agent
- generates a receipt with a random receipt number
- writes receipt number and the encrypted ballot to a receipt list
- encrypts ballot with its own private key
- transmits receipt and signed, encrypted, blinded ballot back to the voter agent

Voter Agent

- uses tallier's public key to confirm that ballot is intact
- unblinds ballot
- transmit encrypted ballot and private key to the tallier

Tallier

- uses private key to decrypt ballot

- writes receipt number and ballot to the receipt list
- transmits an ALL-CLEAR message to the voter agent

Voter Agent

- transmits ALL-CLEAR signal to the validator

Validator

- marks record as having been validated
- writes record to disk
- releases record back to database

Assumptions and Trade-offs

There are some assumptions we made when designing this system. The first assumption is that there is only one remote voting poll site, per county. Since voters now have the ability to access their appropriate ballot from any county within the state, there is no need to have multiple poll sites within one county. Hence we are justified in having each county possess a local copy of the voter registration rolls for each county. Another assumption is that this system is intended to be implemented statewide only. We are not interested in having a voter from California being able to vote in, say, Maine. Today's election systems are administered by states and counties, so our system will also assume the same. Until elections start being administered nationwide, we need not concern ourselves with state-to-state voting at this time.

We decided that the machines used to implement this system should be built specifically for this design in mind. These machines should be built with only the requirements needed for this system; however, there should still be a way for functionality to be added to the machines in the event that other systems can be

implemented on them. Our original idea was to have machines that were used for election then donated or loaned to maybe schools; however, we came to the realization that if we did that, time would be wasted in making the machines secure again and removing any unwanted things on the machines. The condition of the machines could be compromised, especially in a school environment. Also, loaning the machines to another organization would mean that the machines would have to be equipped for higher functionality, which we want to avoid for security purposes.

There are some risks that we perceived but could not find a good enough solution. For one, there is a possibility that a voter's vote may not be counted, even with confirmation from the validator. We could not resolve the problem of being able use the receipt issued to prove how a voter voted. We also found in the Sensus documentation that there is a small chance that two or more voters submit the same encrypted ballot; in that event only one of them will be counted. This can happen due to the hash function and table that are used in the Sensus protocol.

A concern of ours was a design issue listed in Spread's documentation. It stated that Spread does not support very large groups; we weren't sure how to interpret very large and if it the size of the distributed network would effect efficiency and security.

We grappled with the idea of having the Sensus protocol be implemented in just one machine or have separate machines for each component of the protocol. We decided to have separate machines because it adds security in the sense that it would be difficult to trace ballot to voter and also if one component was compromised, the whole machine wouldn't have to be removed from operation, as would happen with just one machine. Instead, the problem component can be replaced with full functional one.

Another thing that we had trouble deciding is where in the Sensus protocol to tell the validator to release the voter's record. We debated on whether or not to have the tallier or the voter agent provide this service and we decided in the end to add that functionality to the voter agent. The validator will have multiple records on lock and the faster that it can release these records, the more records it can take it. We also realized that if there was some system failure, this acknowledgement of the voter's vote would not prevent the vote from being counted.