E-voting System:
Specification and Design Document

March 6, 2003

Jamie Brown
Domari Dickinson
Carl Steinebach
Jeff Zhang

Introduction

During the 2000 General Elections, America realized that our election process is not perfect. To some people, the use of technology will solve all the problems, while others realized that elections could only be improved with technology. Elections are unlike any other transactional event. The result of a national election can have so much at stake, from money, power, to dreams or even lives. While technology can improve our election process, there needs to also exist an improvement in election policy. The policy improvements will help bridge the gap where technology will fall short.

Purpose & Scope

This specification design document will detail the design of an electronic voting system for the state of Maryland. This electronic voting system will enable an eligible voter to vote at any polling site statewide during an election period.

Glossary of Terms

CD – Compact Disk
CDR – CD Reader
CDW – CD Writer
Eligible voter – a United States Citizen who meets all of the federal and state election requirements
RV – registration verifier is a poll worker who verifies a voter's registration
ROM – Read Only Media
Memory – any temporary storage
Kiosk – a device with a touch screen and button inputs that resembles an ATM

User characteristics

This electronic voting system will not prevent any eligible voter from correctly, securely and properly casting a ballot during the election period. Users with special assistance and specific needs may require alternative ballots. These alternative ballots that may exist in audio, video or Braille forms are beyond the scope of this document.

Overview from User Point of View

This specification design document will detail the design of an electronic voting system for the state of Maryland. This electronic voting system will enable a valid voter to vote at any polling site statewide during an election period. Every poll site statewide will have identical architectures. A central voter registration database (VRDB) will contain all eligible voter information. A current copy of VRDB must be available at each poll site prior to the election period. An eligible voter, John Smith, presents identification to the poll worker, registration verifier (RV), for authentication. The RV looks up the

identification information of John Smith and verbally verifies all information including address, zip code and county.

The RV next creates a voting session for John Smith. The voting session has two flags: the voter is present and has been authenticated & the voter has successfully cast a vote. Only the first flag is now set for John Smith. Next, the RV prints off John Smith's voter token onto a piece of paper. This voter token is a piece of paper that contains unique token number, ballot ID, timestamp, polling site ID, and RV ID both in human readable format and machine-readable bar code format. John Smith takes the voter token and waits in line for an unoccupied voting kiosk.

Mr. Smith decides to use voting kiosk #4. After closing the curtain, John notices that the touch-screen has instructions to insert the voter token into the small slot in the middle of the Kiosk. John places the voter token into the slot, and the voting kiosk seizes the voter token until John has successfully voted. Another screen appears prompting John to either select a "Tutorial for operating the Kiosk" or to "Start Voting". John selects "Start Voting". The next screen gives John his home precinct number, county, zip code, and party affiliation to verify. Once John has successfully verified his home precinct ballot information, the first ballot choices are given. John successfully votes for a candidate via a touch screen or button input. Once John has completed voting, a verification and summary page is displayed.

At this point, John has the ability to modify any choices he has made. If no choices need to be modified, John selects the "Accept" button. This sends John's ballot to a paper print out for additional visual inspection and a physical audit record. After visually inspecting the printed ballot, John can either "Accept" or "Reject" this ballot. John chooses to "Accept" this ballot. This action causes a large Accept footer to be written on the paper ballot. The screen thanks the user for voting and reminds John to take his voter token with him as his receipt. The voter token now has additional information such as Kiosk ID, timestamp and that a successful vote has been cast. John's voting experience is complete.

Specifications

Software functions	**CreateVoterToken()**
Device	VR Laptop, VR Laptop CDR, VR Laptop CDW
Pre/post conditions	
Data descriptions	Take variables from VRDB, write & sign variables to token
Data relationships	Invoked by authenticated poll worker
Implementation priorities	High

Software functions	**WriteVoterToken()**
Device	VR Laptop, VR printer

Pre/post conditions	
Data descriptions	Flag set when voter receives token
Data relationships	Invoked by CreateVoterToken()
Implementation priorities	High

Software functions	**ReadVoterVerified()**
Device	VR Laptop, one-way
Pre/post conditions	
Data descriptions	Flag set when VR station receives user completed voting
Data relationships	Invoked by WriteVoterToken()
Implementation priorities	High

Software functions	**WriteVoterVerified()**
Device	Kiosk, one-way
Pre/post conditions	
Data descriptions	Signs and transmit flag to VR station once user vote written
Data relationships	Invoked by WriteVoteCD()
Implementation priorities	Medium

Software functions	**BallotPrecinctLookup()**
Device	Kiosk, barcode reader
Pre/post conditions	
Data descriptions	Reads barcode, verify signature on barcode, lookup ballot,
Data relationships	Invokes TestWriteVoteCD() & TestReadVoteCD()
Implementation priorities	High

Software functions	**VoteNow()**
Device	Kiosk, touchscreen
Pre/post conditions	
Data descriptions	Actual ballot and vote choices
Data relationships	Invoked by BallotPrecinctLookup() Invokes PrintBallotSummary()
Implementation priorities	High

Software functions	**PrintBallotSummary()**
Device	Kiosk, printer
Pre/post conditions	Sends ballot data to printer and waits for “accept” or “reject”
Data descriptions	Prints human & machine readable audit trail, with footer
Data relationships	Invoked by “accept” on VoteNow() Invokes WriteVoteCd()

Implementation priorities	High

Software functions	**WriteVoteCd()**
Device	Kiosk, cd device
Pre/post conditions	
Data descriptions	Writes ballot information to CD along with footer
Data relationships	Invoked by PrintBallotSummary() Invokes WriteVoterVerified()
Implementation priorities	High

Software functions	**TestWriteVoteCd()**
Device	Kiosk, cd device
Pre/post conditions	
Data descriptions	Verifies that CD is operational
Data relationships	Invoked by BallotPrecinctLookup()
Implementation priorities	High

Software functions	**TestReadVoteCd()**
Device	Kiosk, cd device
Pre/post conditions	Verifies that TestWriteVoteCd() is correctly written so voter can start voting process
Data descriptions	Verifies that TestWriteVoteCd() is correctly written
Data relationships	Invoked by TestWriteVoteCd()
Implementation priorities	High

Software functions	**VoteTutorial()**
Device	Kiosk
Pre/post conditions	None
Data descriptions	A sample ballot is given to understand and try Kiosk
Data relationships	Invokes BallotPrecinctLookup() Invoked by user
Implementation priorities	Low

Software functions	**BallotHelp()**
Device	Kiosk
Pre/post conditions	
Data descriptions	
Data relationships	Invoked by user in VoteNow()
Implementation priorities	Low

Software functions	**SignCd()**
Device	Kiosk, cd device

Pre/post conditions	
Data descriptions	Only completed end of election day.
Data relationships	
Implementation priorities	High

Software functions	**ReadSignCd()**
Device	Tallier, cd device
Pre/post conditions	
Data descriptions	Verifies ballot cd written by authenticated kiosk
Data relationships	
Implementation priorities	High

Software functions	**WriteVoteTotal()**
Device	Tallier
Pre/post conditions	
Data descriptions	Sums all ballot totals
Data relationships	
Implementation priorities	High

Constraints

Elections CAN exist on more than one day.
This system MUST only be used for a state-wide election or smaller.
All poll workers MUST be trained on the systems in which they will assist.
All devices MUST be tamper evident.
A random sampling of devices MUST be put through additional pre & post election testing.
All devices MUST undergo a full re-installation and product upgrade between elections.
All devices MUST be secured while in transit, storage, and in use.

Reliability requirements

In the event of a power loss our system cannot be operational. Some systems do offer internal battery backups for momentarily; however, this does not provide for an extended. In addition, it would create a higher end cost, which is undesirable.

Prior to election day it is required that enough resources such as paper, ink, cds and tokens do exist at each polling site, or within their respected devices.

There needs to exist strict policy that enforces systems basic performance benchmarks. An example is that the Kiosk CD-RW must write 10,000 records without an error.

Maintainability

All vital system operation such as transporting, updating, setting up or taking down systems is a two-person detail. This ensures that not one person can corrupt the system. The two-person detail includes the preloading of ROM onto the devices and removing the ballot cds from the kiosks.

In between elections, the systems must be overhauled and updated to ensure a clean system for the next election. There cannot exist any modulation of parts, which might contain historical data.

The systems must be securely transported and stored in different geographical locations.

All systems must undergo a basic testing procedure prior to an election event. Moreover, a random number of systems must undergo a rigorous testing to ensure security and continuity.

Security Requirements

- An individual not registered to vote must not be able to cast a ballot
    - A voter must not be able to vote more than once
- The privacy of the vote has to be guaranteed during the casting, transfer, reception, collection, and tabulation of votes
- No voter should be able to prove that they voted in a certain way voter
    - None of the participants involved in the voting process (organizers, election officials, trusted third parties, voters, etc) should be able to link a vote to an identifiable

–Εαχη ϖοτε ισ ρεχορδεδ πρεχισελψ ασ τηε ϖοτερ ιντενδεδ

- Εαχη ϖοτερ ισ ενσυρεδ α ∀χλεαν σλατε∀ οφ τηε σψστεμ το ενσυρε εθ υαλιτψ, χονφιδενχε, ανδ μινιμιζε σψστεμ ταμπερινγ
- The outcome of the voting process must correspond to the votes cast
- It should be infeasible to exclude a valid vote from the tabulation, and to validate a non-valid one

- System operations are logged and audited
- The system cannot be re-configured during operation
- Access to voted ballots is prohibited until after the close of polls
- Additional ballots cannot be cast once the polling place has closed
- The system must be open to independent inspection and auditing
- The system is protected against accidental and malicious denial of service attacks

Analyzing our system to ensure security

In the process of creating the design specifications, we also asked several questions that served to analyze the security of our system. Such questions include:

Can a person who is not eligible to vote register?
Although our system is not specifically handling the registration details, by defining the term "eligible voter" we have indicated that our system recognizes that there are certain requirements that must be met in order for an individual to register to vote

Can a person who is not eligible, cast a vote?
A person that is not eligible to vote cannot cast a ballot because they will not be verified by the RV, and hence they will not posses the voter token needed to begin a voting session with a kiosk

Can an illegal vote be introduced?
Due to the cryptographic protocols used, an illegal vote cannot be introduced into the system (without being detected?)

Can a legal vote be modified?
Due to the cryptographic protocols used, an illegal vote cannot be introduced into the system

Can a vote be tied to a voter? Can a voter proved HOW they voted?
This system implements the concept of receipt-freeness so it is not possible to a vote to be tied to a voter, or for a voter to prove how they voted

Can a voter vote more than once in his precinct?
Due to the communication between the kiosk and the RV, it is not possible for a voter to vote more than once in his precinct

Can a voter vote more than once in his state?

Is each voter ensured a clean slate when they vote?
A voter is ensured a clean slate when they vote due to the clearing of memories, and the use of ROM

Is the voter's intent recorded by the voting machine?
By allowing a voter to change his ballot (after viewing the print out) and recording such changes, the voter's intent is captured as well as the final vote

Is the voting machine expected to "check itself"?
Utilizing separate machines for vote recording and vote tallying prevents the kiosks from having to "check itself"

Can an illegal vote be introduced during the vote tallying process?
Due to the cryptographic protocols used, an illegal vote cannot be introduced into the system

Some questions that were raised, and not answered include:

How do we ensure that the vote tallying is accurate?

How do we guarantee that the information in the VRDB is accurate?

How are votes cast in the event that a kiosk fails? What happens when a kiosk fails and causes a ballot not to be cast?

What happens in the event that the registration device fails, causing the bar codes not to be created?

What happens in the event that the printer fails, and a voter cannot view a printed copy of his ballot selections?
What happens when a kiosk fails to retrieve correct precinct ballots?

Architectural Design

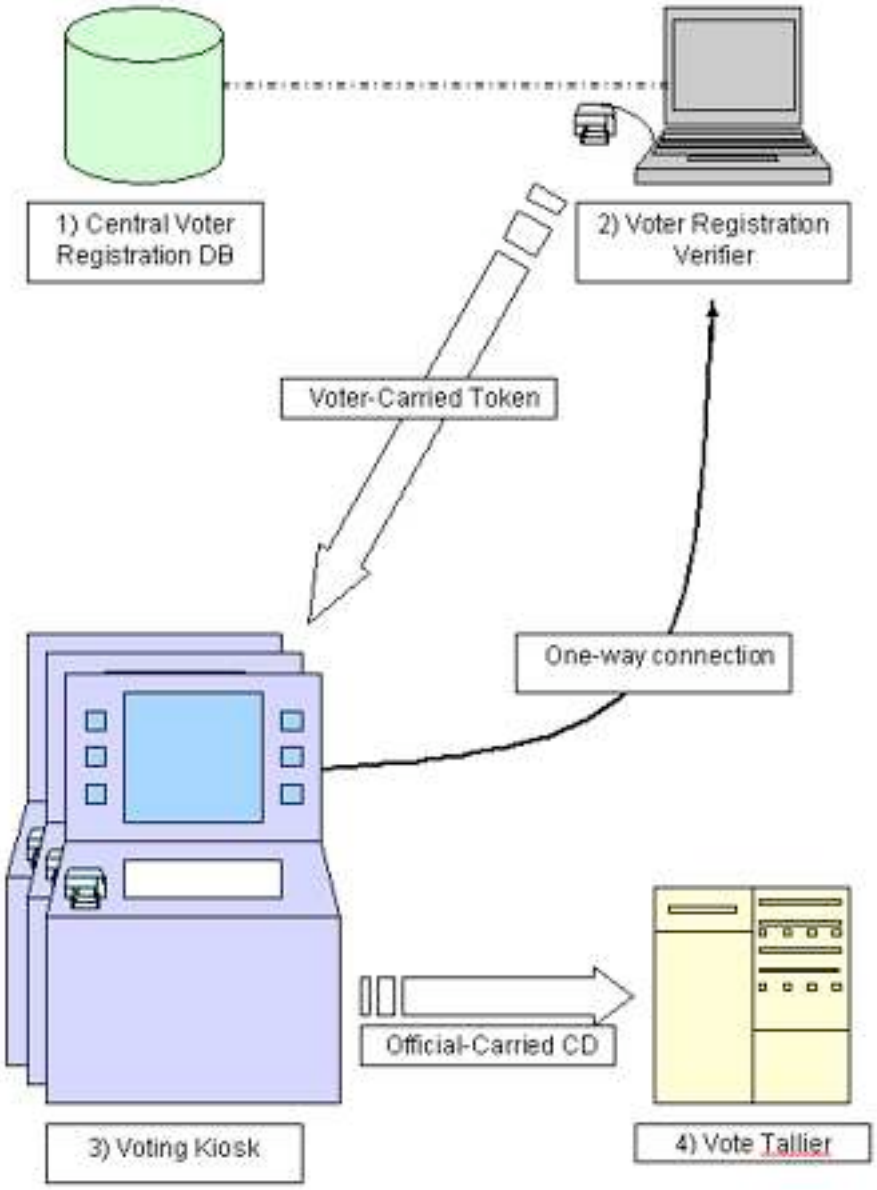


Figure 1 Architecture

Central Voter Registration Database (VRDB)

The VRDB is a state wide master voter registration database. The VRDB has records such as First Name, Last Name, Middle Initial, Suffix, Address1, Address2, City, State, Zip, County, Phone, Voter ID, Party Affiliation, and Precinct ID for each eligible voter.

Only Election Officials and the Motor Vehicle Association have trusted access to the master database.

Registration Verifier (RV)
Every RV has a smart card that enables local copy of the VRDB to be decrypted. The RV has an authenticated local copy of the VRDB. The RV uses the VRDB information to authenticate eligible voters. Once the voter is authenticated a flag is set starting the voter session and a voter token is printed out. This voter token is a piece of paper that contains unique token number, ballot ID, timestamp, polling site ID, RV ID, and RV digital signature both in human readable format and machine-readable bar code format. After the voter has successfully voted, the Voting Kiosk will transmit a "voted" packet to close the voter's session. The RV has an audit record of each voter and voter session.

Voting Kiosk
The voting Kiosk is where all the action is located. To start, the voter must place the voter token into the slot. The voting kiosk will seize this token until the voter has successfully voted. After the token has been seized, the kiosk will verify that this token is valid authentic, this is done by looking at the RV signed token, timestamp and the polling site id. The RV public key is stored in an LDAP directory locally available on a ROM. If these variables are correct, then the ballot id is verified. The ballot id is an identifier that lets correctly selects the voter's home precinct, by listed the voter precinct id, county, zip code and party affiliation. All the ballots for the entire state are pre-loaded on a ROM in each kiosk; this ensures that each new voter will have a "clean" ballot. The ballots will be coded in XML format with ballot specific help formats. After the correct ballot is retrieved, the screen prompts the voter to verify the correct ballot or use the kiosk tutorial. If the ballot listed is incorrect, the kiosk signals a poll worker to assist the voter. If the correct ballot is listed, then the voter can start voting. If the user selects to use the kiosk tutorial, then the voter is taken into a hands-on learning tutorial detailing how to use the kiosk.

During the voting process, the voter can request assistance from a poll worker at any time and there is no time limit on voting. The voter will select a candidate for each office and be able to vote on the issues. After voting is completed, a verification and summary page is prompted. At this point the voter is to double-check their vote choices. If the user needs to make modification, then they can select the office or issue to change. Once, the voter is done with the verification-summary page, the voter selects "Accept". This option will print the verification-summary page to a paper output. Here the voter can visually inspect and verify the correct candidates were chosen. Again, the voter has the ability to modify the ballot. If the voter voted incorrectly, then the printed ballot is printed with "REJECT" footer, signally that this paper ballot has been rejected. The voter is then taken back to the verification-summary page for ballot modification. However, if the voter decided he correctly voted, then the paper ballot footer is printed with an "ACCEPT". The paper ballot footer also has the timestamp, kiosk id and polling site id.

The voting kiosk uses a randomly-generated 128-bit session key to encrypt the ballot data using AES in CBC mode using 128-bit blocks. This session key is then wrapped with the

tallier's public key using RSA. Finally, the entire package is digitally signed with the voting kiosk private key. Now, the actual vote is cast by writing the ballot to the CD from memory. This hardware separation is required to prevent the loss of an entire kiosk's ballots. Moreover, the hardware separation will aid in tallying the votes by having removable media.

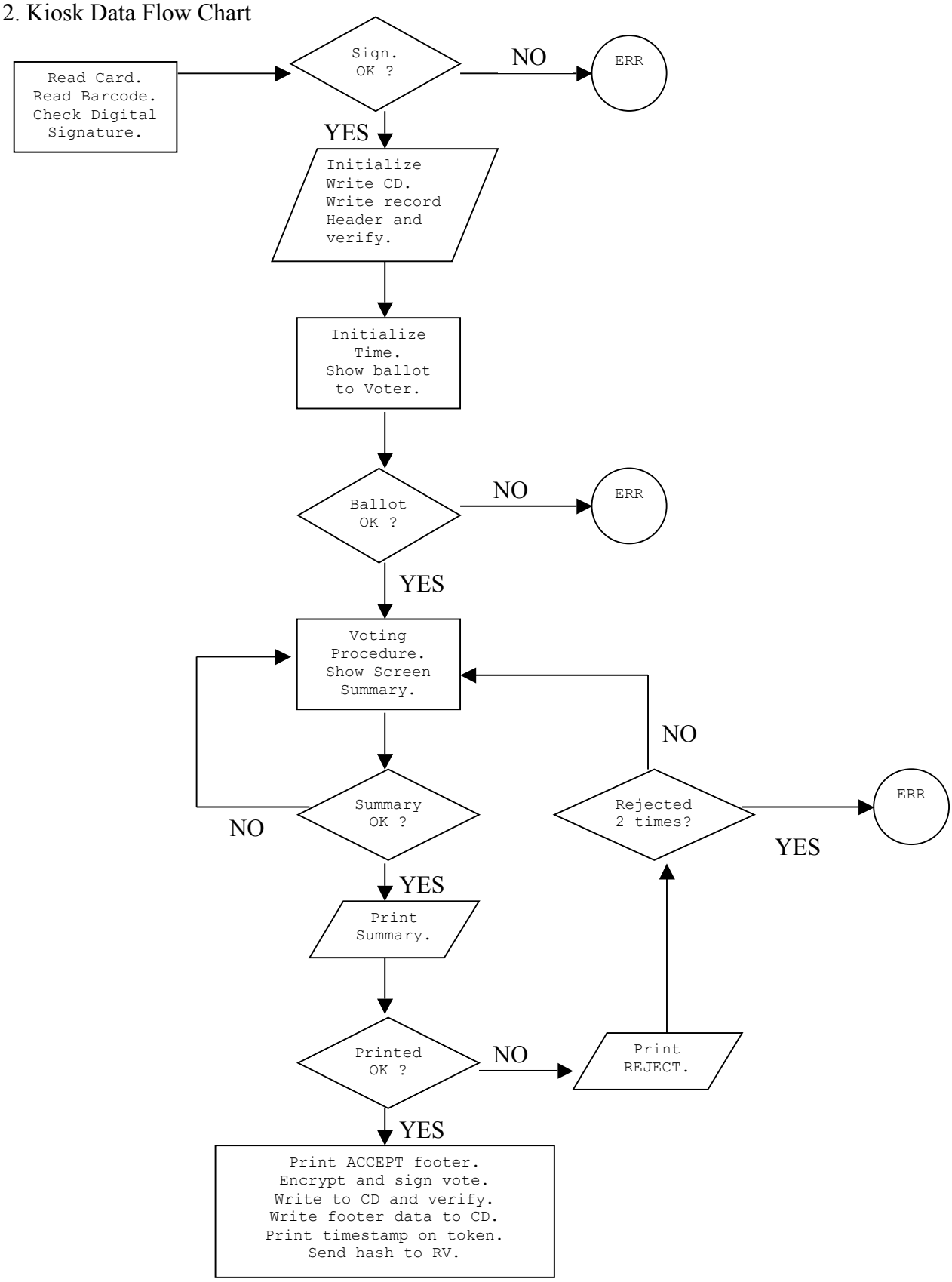
After the paper ballot has the "ACCEPT" footer, then the voter token is written with the same footer. Once the voter token has the footer written on it, the token is invalid, which causes the voting kiosk to discharge it from the slot. The voter token is now a physical receipt for the voter, given evidence of voting time, location and kiosk information. This receipt does not link the voter back to the ballot in any way. During the writing of the footer on the token, another crucial step is happening within the voting kiosk. The kiosk sends the same footer to the VR via a one-way network connection, signaling that the voter has successfully voted. This will close the voter session on the VR. This step will provide the much-needed accountability of number of voters entered, sessions, ballots, and votes cast. The one-way connection will ensure that the VR cannot leak information about the voter to the voting kiosk or to any cast ballots.

After the polling site is closed, the kiosk digitally signs the ballot cd and fills a remaining space with a unique sequence to ensure data integrity.

Vote Tallier

A vote tallier is centrally located at each county within a state. At the end of the Election Day, the Ballot CDs are removed from the voting kiosks by a trusted election official. The election official takes the Ballot CDs to a central county location for an official vote tally. Prior to tallying, each Ballot CD must be verified that it came from a trusted and authentic Kiosk.

Figure 2. Kiosk Data Flow Chart



Detailed Design

Module Name	**Central Voter Registration Database**
Description	A state-wide master voter registration database
Interface Specification	Election Officials & Motor Vehicle Association
Process Description	Master database closes for new entries or modifications 30 days prior to election day.
Initialization Requirements	
Exception Handling	

Module Name	**Registration Verifier**
Description	Authenticates voter via a local copy of voter registration database and token writer
Interface Specification	
Process Description	
Initialization Requirements	The local copy of the VRDB must be installed onto the system. Smart card loaded with VRDB public keys for decrypting
Exception Handling	

Module Name	**Voting Kiosk**
Description	Reads token, actually stores and casts ballot
Interface Specification	See Figure 2 flow chart
Process Description	
Initialization Requirements	All the ballots statewide must be loaded into ROM. All RV public keys stored in LDAP must be loaded into ROM The printer must be refilled on paper and ink. The ballot cd must be in place in the tamper evident case The kiosk smart card must be loaded in tamper evident slot. All tallier public keys
Exception Handling	

Module Name	**Vote Tallier**
Description	Authenticates ballot cd and sums the vote totals
Interface Specification	
Process Description	
Initialization Requirements	All voting kiosk public keys
Exception Handling	

Alternative solutions

Title	Human-in-loop
Description	After casting a vote, have the ballot on the receipt. This would require the voter to walk over with the receipt to a tallying machine.
Reason for rejection	This requires the human voter to successfully hand in a receipt to the tallying machine. What if the human forgot, or purposely failed to submit the receipt? This option caused much confusion and a mess of other issues such as buying votes.

Title	Official published list for candidates
Description	After the official results are certified, they are published online in detail. This would allow me to know that I voted on ballot number 123. I would be able to look at the candidate Bush and know that ballot 123 counted for one of his 2 million votes. I will know that my vote was correctly counted.
Reason for rejection	The main reason for rejection is what if I really voted for Gore. This official result is published weeks after the election, what recourse do I have as a voter?

Title	Voting Kiosk and Vote Tallier within same machines
Description	We wanted machine separation so that we could have checks and balances on the machines. We did not want the kiosk to both create and then count the vote. We needed the voting kiosk do a job and then have the tallier do a job.
Reason for rejection	Even though it adds complications to the design, it is better to have role separation.

Title	Solve one vote per person state-wide
Description	We are only able to solve this for each precinct locally. This would require database synchronization every time a voter is verified and then successfully voted.
Reason for rejection	The problem quickly becomes overwhelming and would cause us to have the RV networked to the VRDB. What if a database was out of sync? Which one is correct?

Title	Link Voter to Ballot
Description	Ballot secrecy is crucial to instilling voter confidence. We attempted in every manner to restrict access to the

	VRDB and include separate machine roles. Moreover, also include the one-way connection from the kiosk to the RV.
Reason for rejection	Mainly, voter confidence.

Title	Networked machines
Description	We wanted physical separation to ensure less chance of data corruption across systems.
Reason for rejection	This architecture opens up more issues.

Tradeoffs

In finalizing the design specifications for this project, this group was forced to consider and weigh a variety of tradeoffs. They include usability vs. efficiency, receipt-freeness vs. verifiability, privacy vs. verifiability, security vs. cost, networked machines vs. stand alone machines, and using a hard drive vs. ROM.

In order for this electronic voting scheme to be successful, in addition to carrying out the basic election tasks, it must also be as easy to use as the current voting schemes. If the proposed scheme is a drastic change from what voters are used to, they may be deterred and refrain from voting. Keeping this in mind, this group chose to concentrate on the system's usability rather than its efficiency. This is not to say that the system should be considered inefficient, but there are some aspects that can be implemented in a manner that may be considered more efficient. The reason that we chose our particular design is that our system strives to provide a positive voting experience. For example, the ballot layout that we chose requires that each candidate or measure appear on its own separate screen. From a programmer's point of view, it would be more efficient to have as many possible candidates on a single screen, but this may cause the voter to become confused and create errors in his ballot selections. Also, allowing the voter to view his ballot via a paper print out, and accept or reject this ballot may seem to be an unnecessary step that only adds to the duties of the system, but we chose to include this for several reasons. Not only does this step help ensure the accuracy of the ballots cast, but it also adds to voter confidence by giving the voter some indication that his choices were recorded by the machine.

Another issue that was encountered was that of receipt-freeness vs. verifiability. When trying to decide between the two, we originally focused on ensuring that the voter had some way of knowing that his vote counted. However, devising a secure scheme such that this was possible was very difficult, so we focused on implementing the idea of receipt-freeness. It is important to note that although our system provides the voter with a voter token once they have completed the voting process, this token does not invite corruption as do receipts by introducing avenues for vote selling/buying, coercion, or fraud.

As mentioned earlier, utilizing technology can improve the voting process, but any voting scheme implemented should ideally increase voter confidence, and it must ensure the

integrity of the election process. For these reasons a voter must be ensured that no unauthorized individual can access his personal information or ballot selections, and that these two entities will remain separated. This group chose to focus more on the privacy aspect of the system then on verifiability due to the same implementation issues mentioned above.

Once our entire system was designed, we realized that to actually implement such a system would be very, very expensive. Between the kiosks, laptops, printers, paper, encryption tools, and personnel, implementing this e-voting scheme would require the state of Maryland to invest a significant amount of money in support of improving the current voting techniques. Instead of compromising the security of our design by making modifications in an attempt to reduce the overall cost, this group believes that the security and overall effectiveness of the system is more important than its cost.

The decision of stand-alone machines not having a network connection is crucial to our design. A network connection does aid in efficiency, by not requiring machines to have pre-loaded datasets. However, the main reason for stand-alone machines was precisely the reason of usability. There exist 1600+ precincts in Maryland, it is not practical to have 4 more kiosks at each precinct to query a central database for particular ballots thousands of times a day. It is more efficient in voter time to have as much stored locally as possible. This also prevents many attacks that have been sited as concerns in many reports.

The important tradeoff of using memory device cards like smart cards or hotel room keys as voter tokens vs a paper bar code helped work out some potential usability errors. We realized that smart cards or even hotel room keys are more expensive than simple paper. Since cost is a real issue, we realized that most sights would “recycle” these tokens. This could create a massive mix up if not done correctly. Even still, when done correctly, there exists a potential for data leakage. Finally, a positive outcome from our decision was the fact that a receipt is now capable. The MIT paper suggests that all tokens be seized after use, but this would never happen in practice.

Lastly, the tradeoff between a Kiosk hard drive and a ROM for storing the ballots and LDAP public keys for the tallying machines is to ensure a “clean” ballot for each voter. While, it is possible to have a read-only hard drive, it is not cost effective or practical for creating thousands of them. Finally, the hard drive is just over kill for the space requirements. The ballots are in text file XML format.

References

Voting What Is What Could Be. MIT & Caltech. July 2001.
http://csrc.nist.gov/
www.datakey.com for smart cards
www.rsasecurity.com for RSA
http://www.free-project.org
http://www.blackboxvoting.com/
http://mainline.brynmawr.edu/~rmercuri/notable/evote.html

http://www.elections.state.md.us/
http://www.mdvotes.org/
http://www.elections.state.md.us/registered_voters/index.html
http://www.spectrum.ieee.org/WEBONLY/publicfeature/oct02/evot.html
www.votewatch.us "A repository for Voter Complaints"
http://firstgov.gov/Citizen/Topics/Voting.shtml
http://www.ecotalk.org/VotingSecurity.htm