

InnoVote SecureDRE

Functional Design

By
Erin Thead
Software Engineer
erin@erinthead.com

© 2005

Table of Contents – SecureDRE Functional Design

1.	Introduction.....	201
1.1.	Purpose.....	201
1.2.	Scope.....	201
1.3.	Definitions, Acronyms, and Abbreviations.	Error! Bookmark not defined.
1.4.	References.....	202
1.5.	Overview.....	202
2.	Overall Description.....	203
2.1.	Product Functions.	203
2.2.	User Classes.....	203
2.2.1.	Voter user class	203
2.2.2.	Election Official user class	204
2.2.3.	Software user	204
2.3.	Assumptions and Dependencies.	205
2.3.1.	Hardware platform assumption.....	205
2.3.2.	Internet connectivity assumption	205
2.3.3.	Single-user assumption	205
2.3.4.	Precinct computer assumption	205
2.4.	Deployment of the Software.....	206
3.	Specific Requirements	207
3.1.	Functional Requirements.....	207
3.1.1.	System Feature 1: Enter private network.....	207
3.1.2.	System Feature 2: Program the machine	208
3.1.3.	System Feature 3: Display choices to Voter	209
3.1.4.	System Feature 4: Accept a user’s vote	210
3.1.5.	System Feature 5: Display current votes	210
3.1.6.	System Feature 6: Modify choice	211
3.1.7.	System Feature 7: Cast ballot	212
3.1.8.	System Feature 8: Print ballot.....	213
3.1.9.	System Feature 9: Change printed ballot	213
3.1.10.	System Feature 10: Send real-time election results	215
3.1.11.	System Feature 11: Finalize vote.....	216
3.1.12.	System Feature 12: Lock voting machine.....	216
3.1.13.	System Feature 13: Unlock voting machine	217
3.1.14.	System Feature 14: End election.....	217
3.1.15.	System Feature 15: Send finalized election results.....	218
3.1.16.	System Feature 16: Detect and log errors	219
3.1.17.	System Feature 17: Clear election results	220
3.2.	Performance Requirements.....	221
3.2.1.	Performance Requirement 1: Modify Database quickly.....	221
3.2.2.	Performance Requirement 2: Print ballot quickly	221
3.2.3.	Performance Requirement 3: Transmit and receive data quickly	221

3.3.	Security Requirements.....	222
3.3.1.	Security Feature 1: Limit input from keyboard	222
3.3.2.	Security Feature 2: Verify identity of data transmitters.....	222
3.3.3.	Security Feature 3: Encrypt Database tables	223
3.3.4.	Security Feature 4: Restrict data flow to Database tables.....	223
3.3.5.	Security Feature 5: Limit changes to Database on Election Day.....	224
3.3.6.	Security Feature 6: Private network.....	224
3.3.7.	Security Feature 7: Encrypt outbound data.....	225
3.3.8.	Security Feature 8: Block all ports except two	225
3.3.9.	Security Feature 9: Database login	225
3.4.	System Attributes.....	226
3.4.1.	Reliability.....	226
3.4.2.	Availability	226
3.4.3.	Security	226
3.4.4.	Maintainability	226
3.4.5.	Portability.....	226

1. Introduction

1.1. Purpose.

The purpose of this document is to communicate the software requirements and functional design for the InnoVote SecureDRE System. The document provides a detailed description of functional, performance, and security requirements, design constraints, and classes of persons who will be using the software.

This document does not address hardware-operation and hardware-management issues except as they relate to a software operation related to conducting an election. Reference [4], *MyVotronic Hardware and Operating System Overview*, addresses such low-level hardware operation details.

The intended audience of this document is the developer and any other persons interested in the project, including election reform activists, computer security professionals, political figures with an interest in election reform, and potential buyers of the design.

1.2. Scope.

The InnoVote SecureDRE System is one component of an interoperable line of products. It is a software product designed to execute on the proposed “MyVotronic” Direct Recording Electronic voting machine [ref. 4].

SecureDRE will manipulate data received from the MyVotronic hardware as well as from other InnoVote products as necessary. The software will provide a GUI to voters to allow them to interact with it and perform actions that will generate data for SecureDRE to process.

Current DRE voting machine technology is demonstrably insecure and unreliable. Existing DRE machines, which are used in voting sites across the United States, have had numerous and severe documented errors, including the following:

- Vote tallies accidentally set to negative numbers
- Votes being counted for the wrong candidate
- Votes being electronically generated in error
- The suspicion of pre-election rigging of machines
- Unverified updates being made to election software
- Public online access to sensitive operations of election software

In contrast to most Direct Recording Electronic voting equipment software, SecureDRE is a highly secure and accountable system with strong anti-fraud protection. Election data are heavily protected both from errors and from tampering. In the unlikely event that data stored on a MyVotronic machine *are* compromised beyond recall, SecureDRE provides a physical backup of election data in the form of paper ballots.

1.3. References.

- [1] Thead, E. *InnoVote CardReader Hardware Requirements Overview*, 2005.
- [2] Thead, E. *InnoVote CardReader Functional Design*, 2005.
- [3] Thead, E. *InnoVote Database Detailed Design*, 2005.
- [4] Thead, E. *InnoVote MyVotronic Hardware and Operating System Overview*, 2005.
- [5] Thead, E. *InnoVote Network Detailed Design*, 2005.
- [6] Thead, E. *InnoVote ReliaVote Central Server Functional Design*, 2005.
- [7] Thead, E. *InnoVote ReliaVote Precinct Edition Functional Design*, 2005.
- [8] Thead, E. *InnoVote Database Access Matrix*, 2005.
- [9] Thead, E. *Security Analysis of InnoVote Products*, 2005.

1.4. Overview.

The remainder of this document is organized in the following fashion:

Section 2: Provides overall description of the system, including product functions, user classes, assumptions, and generalized dependencies.

Section 3: Provides specific requirements including functional requirements, performance requirements, and security requirements.

Section 4: Provides supporting figures and tables for information in other sections of the document.

2. Overall Description

2.1. Product Functions.

SecureDRE will need to perform the following basic functions:

1. Accept data transfers from the ReliaVote Precinct Edition software product "programming" the machine. This involves adding and/or modifying entries in Database tables.
2. Display candidates and options for contests on the ballot
3. Allow users to vote for displayed candidates and ballot options
4. Display a list of the voter's current choices for all issues on the ballot
5. Allow the voter to modify his/her choices before the ballot is cast
6. Print a paper copy of a voter's ballot
7. Allow the voter to change his/her ballot even after it has been printed
8. Finalize a vote and lock it from further changes as soon as another vote is cast
9. Transmit votes in real-time to the ReliaVote Precinct Edition software
10. Accept a signal from ReliaVote Precinct Edition indicating that the election is ended
11. Lock input from and output to hardware when an election is over
12. Unlock hardware to allow input and output
13. Send the final vote results and tallies to the precinct computer
14. Allow write-in candidates where applicable
15. Maintain error logs for itself and receive error logs from voting equipment in the network

2.2. User Classes.

2.2.1. Voter user class

The Voter user class represents a citizen of the United States who is casting a ballot in a primary or general election. This class of user will need to perform operations that modify sensitive tables in the InnoVote Databases.

The Voter user class is assigned to all input from the MyVotronic hardware until the Operating System assigns it a different privilege level.

Members of the Voter class must be allowed to perform operations that will do the following:

- Add entries to the Votes and Realtime_Votes tables in the Database
- Modify entries in those tables that they have created, but only those

- Modify the Tallies table

It is *not* necessary for these operations to be performed at the Voter privilege level. The operations can be initiated by a Voter and the changes to the Database can be performed by the Operating System.

2.2.2. Election Official user class

The Election Official user class represents a citizen of the United States who has the legal authority to oversee a general or primary election in a precinct. This class of user will need to initiate operations that modify sensitive tables in the InnoVote Database but should *not* have direct access to the Database.

Election Official privileges cannot be gained directly from the MyVotronic or SecureDRE. This user class is assigned to any incoming data transfers that originated from ReliaVote PE until the Operating System assigns a different privilege level. It is assumed that only election officials or persons acting under the legal authority of election officials will be using the ReliaVote software.

The Election Official does not have direct access to the Database. Any Election Official requests for modifications to the database will be performed at the Operating System privilege level.

2.2.3. Software user

The Software shall function as a user class. It must be able to perform any operation necessary on the hardware and data stored in the system. Operations in software functions initiated by Voters or Election Officials may be executed at Operating System level if necessary. These functions are described in §3 of this document.

The Software will in theory have full access to the Database at all times, but it must be coded so that it will in *practice* not perform certain damaging operations to the Database, and that no human user or external machine will be able to interact with a MyVotronic with Operating System privileges.

2.3. Assumptions and Dependencies.

2.3.1. Hardware platform assumption

The document assumes that InnoVote SecureDRE will operate on the electronic voting machine hardware “MyVotronic” or compatible hardware, with “MyVotronic OS” or compatible operating system. SecureDRE has numerous functions that require certain hardware features for correct operation.

2.3.2. Internet connectivity assumption

The document assumes that hardware running InnoVote SecureDRE will be connected to a TCP/IP-compatible network during operation. IP version 6 is not necessary for correct operation of SecureDRE, but it does provide extra security benefits.

2.3.3. Single-user assumption

The document assumes that SecureDRE is a single-user software product. Although there is no provision for user profiles and logins, it is conceivable that SecureDRE could receive instructions from more than one concurrent source or “user,” as defined by the User Classes (§2.2) in this document.

For any set of operations that could potentially be in conflict, the operation currently being executed by the Software user has precedence. Next are operations that were initiated by Election Officials. Lowest in precedence are operations that were initiated by Voters.

2.3.4. Precinct computer assumption

The document assumes that a MyVotronic voting machine running SecureDRE will be connected over a network to one and only one computer that is running “InnoVote ReliaVote Precinct Edition” or compatible software. Several operations of the SecureDRE product require receiving data from ReliaVote Precinct Edition for correct operation.

3. Specific Requirements

3.1. Functional Requirements.

The features described in this section are operations that are necessary for correct and useful operation of the SecureDRE software product.

3.1.1. System Feature 1: Enter private network

3.1.1.1. Purpose of Feature

This feature allows the Software to establish a unique IP address on a precinct-level private network, open a data-transfer port for SecureDRE software to use, and identify itself to the precinct computer. The precinct computer can then maintain a table of all devices that are located on the network.

3.1.1.2. Stimulus-Response Sequence

1. The Software scans through the machine's data transfer ports and determines whether any of them are currently open other than a single port that is used for communication with the precinct Kerberos server. Conditional: If any other port is open by another application, the software creates an entry in the Events log of the Database containing the name of the service and the port number in use. When it reaches the last port, the Software pauses execution and displays a message to the user containing any open port numbers and the applications using them, informing the user that it will not continue execution until the ports are closed.
2. The System determines a single data-transfer port to use on the network. This port should be the same for all installations of SecureDRE.
3. The System receives a data transmission from the precinct computer, which has recognized the machine's presence on the network.
4. The System accesses the key management server and verifies the data's encryption keys.
5. The System decrypts the data and determines them to be requests from the precinct computer to for its type of equipment, its hardware version, its software version, and its current status.
6. The System transmits the requested information to the precinct computer.
7. The System records an entry in its Events table indicating the event.

3.1.1.3. Dependencies

This operation requires the completion of no other System Features before it can begin execution.

This feature is restricted by Security Feature 8, "Block all ports except two." The feature requires Security Features 2, "Verify identity of data transmitters," and 7, "Encrypt outbound data."

3.1.2. System Feature 2: Program the machine

3.1.2.1. Purpose of Feature

This feature allows the system's Database to be modified so that the appropriate races, candidates, and candidate information will be displayed to the Voter. In a traditional context, this is called "programming" the voting equipment.

3.1.2.2. Stimulus-Response Sequence

1. The system receives one or more data transmissions from the precinct computer.
2. The system verifies the data's encryption keys.
3. The system decrypts the data and determines them to be instructions to add, delete, or modify entries in the Candidates, Parties, Contests, Running, and/or Affiliations tables in its Database.
4. The system attempts to make the specified changes to the correct tables in the Database.
5. The system records an entry in its Events table indicating the success or failure of the operation.
6. The system sends a notification to the precinct computer of its success or failure to make the changes to its database. Conditional: If the machine failed to perform an operation, the notification contains the error code and error text.

3.1.2.3. Dependencies

This operation requires the completion of System Feature 1, "Send identification to precinct computer," before it can begin execution.

This feature is restricted by Security Features 5, "Limit changes to Database on Election Day," and 8, "Block all ports except two." The operation requires Security Features 2, "Verify identity of data transmitters," 3, "Encrypt Database tables," 7, "Encrypt outbound data," and 9, "Database login."

3.1.3. System Feature 3: Display choices to Voter

3.1.3.1. Purpose of Feature

This feature allows the general ballot information stored in the system's Database to be displayed to a Voter. For each race on the ballot, it displays an option of "No Vote," allowing a Voter to abstain from voting in that race. The system does not advance to the next contest until the Voter has made a choice.

3.1.3.2. Stimulus-Response Sequence

1. A Voter indicates to the system that he/she wishes to begin voting.
2. The system generates a temporary ballot in memory that contains the Voter's choices until the vote is cast.
3. The system queries the Database to determine the races and choices to display to the Voter.
4. The system reads the candidate/contest combination listed in the Running table for the first contest.
5. The system displays a listing of all candidates or options for a particular race on-screen and allows the Voter to view detailed information about each one.
6. The Voter selects a choice for which to view in-depth information.
7. The system queries the database for party affiliation, candidate information, and contest information associated with the Voter's selection.
8. The system displays all information discovered in Steps 3 and 4 to the Voter. Whenever this step is executed, the system also displays an option to allow the Voter to abstain from voting in the race for which choices are being displayed.
9. The Voter chooses either to vote for the choice being displayed or to view the next option for that particular race.
10. The system's operation depends on the Voter's behavior after Step 9.
 - a. If the Voter chooses to vote for the choice displayed during Step 8, the the system executes System Feature "Accept a user's vote" (§3.1.4), returns to execution of this Feature, queries the database for the first Running table entry for a different contest than the one just voted on, and reverts to Step 5 for the new contest.
 - b. If the Voter chooses to view the next choice, the system displays the next entry in the Running for the contest currently being viewed. Exception: If all choices for a given race have been viewed, the system displays the first choice again and displays a message that all candidates have been viewed. The system loops through all choices as many times as the Voter requests.
11. The system records an entry in its Events table indicating the success or failure of the operation.

3.1.3.3. Dependencies

This feature requires the successful completion of System Feature 2, "Program the machine," before it can begin execution.

This feature requires Security Features 1, "Limit input from keyboard," and 9, "Database login."

3.1.4. System Feature 4: Accept a user's vote

3.1.4.1. Purpose of Feature

This feature allows a Voter to vote for one choice or to choose to abstain from voting for a particular race on the ballot.

3.1.4.2. Stimulus-Response Sequence

1. The Voter presses his/her choice on the touchscreen.
2. The system reads the touchscreen input and matches it with the correct choice in the Database.
3. Conditional: If the Voter has selected "write-in" for a particular contest, the system unlocks the keyboard and allows the user to input the write-in candidate's name.
4. Conditional: If the Voter has selected "write-in" for a particular contest and completed Step 3, the system reads the user's input and stores it.
5. Conditional: If the Voter has selected "write-in" for a particular contest and completed Step 4, the system locks the keyboard from input.
6. The system changes the temporary ballot generated in System Feature 3, Step 2, to reflect the Voter's choice.
7. The system records an entry in its Events table indicating the success or failure of the operation.

3.1.4.3. Dependencies

This feature requires the successful completion of System Feature 3, "Display choices to Voter," before it can begin execution.

This feature requires Security Feature 1, "Limit input from keyboard."

3.1.5. System Feature 5: Display current votes

3.1.5.1. Purpose of Feature

This feature allows a Voter's current choices to be displayed before his/her vote is actually cast. This operation is a necessary operation to take in modifying a vote.

3.1.5.2. Stimulus-Response Sequence

1. At any time while a Voter is using the SecureDRE software, the system is displaying an option to "View Current Choices" on-screen.
2. The Voter selects this option.
3. The system displays a list of all races with the Voter's current choice for each race selected. The individual races can be selected by the Voter, an action that initiates the "Modify choice" operation. The system also displays a mechanism to cancel the view.
4. The system's operation depends on the Voter's behavior after Step 3.

- a. If Voter chooses to cancel, the system returns to the operation that was being performed before the Voter initiated the “Display current votes” operation.
 - b. If Voter chooses to modify a choice, the system performs the “Modify choice” operation (§3.1.6) and exit this operation.
5. The system records an entry in its Events table indicating the success or failure of the operation.

3.1.5.3. Dependencies

This feature requires the successful completion of System Feature 2, “Program the machine,” before it can begin execution. It does *not* require that a Voter make a selection.

3.1.6. System Feature 6: Modify choice

3.1.6.1. Purpose of Feature

This feature allows a Voter to modify his/her vote for a race before the ballot is electronically cast.

3.1.6.2. Stimulus-Response Sequence

1. The Voter selects a particular race on his/her electronic ballot.
2. The system displays the Voter’s selected race with the Voter’s current choice selected and mechanisms for modifying that choice, viewing the next race and selection, and canceling.
3. The system’s operation depends on the Voter’s behavior after Step 2.
 - a. If Voter chooses to view the next race, the system displays the Voter’s selection for the next race on the ballot, with same mechanisms present on-screen as in Step 2.
 - b. If Voter chooses to cancel viewing his/her choices, the system reverts to the operation that was being performed before the Voter initiated the “Modify choice” operation and exits this sequence of steps.
 - c. If Voter chooses to modify a choice, the system performs Step 4.
4. The system performs Steps 3 – 10 of System Feature 3 (“Display choices to Voter”) and returns to this operation.
5. The system updates the temporary ballot to reflect the Voter’s choice from Step 4.
6. The system records an entry in its Events table indicating the success or failure of the operation.
7. The system exits this sequence of steps and performs Step 3 of System Feature 5 (“Display current votes”).

3.1.6.3. Dependencies

This feature requires the successful completion of System Feature 5, “Display current votes,” or Steps 1 – 6 of System Feature 9, “Change printed ballot,” before it can begin execution.

3.1.7. System Feature 7: Cast ballot

3.1.7.1. Purpose of Feature

This feature allows the system to electronically record the Voter's choices in the system Database for retrieval later by tabulation software.

SecureDRE generates ballot identification numbers using a pseudorandom incremental generator. Ballots cast later have higher ballot ID numbers than earlier ballots. An out-of-place ballot would indicate a fraudulent entry or system error. SecureDRE then appends an alphabetical code to the beginning of the number that identifies the ballot as coming from a particular machine.

3.1.7.2. Stimulus-Response Sequence

1. The Voter chooses some ballot option, including abstention, for all contests in the Contests table [ref. 3] in the Database.
2. The system recognizes that the Voter has made a selection for all contests and displays an option to the Voter to cast the ballot.
3. The Voter chooses to cast the completed ballot and selects this option.
4. The system performs the operations of System Feature 5, "Display current votes."
5. The system's operation depends on the Voter's behavior after Step 3.
 - a. If Voter chooses to modify one or more selections, the system performs the steps in System Feature 6, "Modify choice," and returns to the beginning of this feature after completion of that operation.
 - b. If Voter chooses to cast the ballot, the system performs Step 6.
6. The system generates a new unique entry in the Ballots table with archival and ballot-lock attributes both set to "FALSE."
7. The system creates new entries in the Database in the Votes and Realtime_Votes tables containing the Voter's choices, with the ballot identification matched to the ballot identification of the new entry in the Ballots table.
8. The system performs the operations in System Feature 8, "Print ballot," and returns to this feature after completion of that operation.
9. The system increments (by one) the values in the Tallies table for the candidates that the Voter selected.
10. The system performs the operations in System Feature 10, "Send real-time election results."
11. The system records an entry in its Events table indicating the success or failure of the operation and exits this sequence of steps.

3.1.7.3. Dependencies

This feature requires the successful completion of System Feature 4, "Accept a user's vote," for every contest in the Database before it can begin execution.

This feature requires Security Features 3, "Encrypt Database tables," 7, "Encrypt outbound data," and 9, "Database login."

3.1.8. System Feature 8: Print ballot

3.1.8.1. Purpose of Feature

This feature allows the system to print a modifiable paper ballot displaying a Voter's choices for every race on the ballot. This ballot is compatible with paper-ballot-scanning equipment.

3.1.8.2. Stimulus-Response Sequence

1. The system generates a human-readable ballot containing the Voter's choices for each race. This ballot has a bar code that is recognizable in digital form by the Database as the ballot identification attribute in the Ballots table.
2. The system prints the generated ballot and instructs the Voter to review it for accuracy.
3. The system records an entry in its Events table indicating the success or failure of the operation.

3.1.8.3. Dependencies

This feature requires the successful completion of System Feature 7, "Cast ballot," or System Feature 9, "Change printed ballot," before it can begin execution.

This feature requires Security Features 3, "Encrypt Database tables," and 9, "Database login."

3.1.9. System Feature 9: Change printed ballot

3.1.9.1. Purpose of Feature

This feature allows a Voter to change his/her vote even after a paper ballot has been printed. By law, areas that use paper ballots must provide a voter with a replacement ballot upon request (and presentation of the original ballot). This feature allows for equal protection for voters who use electronic voting machines. The feature determines whether a ballot has been locked from further modification before making any changes to it. This prevents anyone from making fraudulent changes to a ballot after the Voter who cast it has left the area.

3.1.9.2. Stimulus-Response Sequence

1. The Voter scans the printed ballot's bar code with the MyVotronic machine's bar code scanner.
2. The system reads the input from the bar code scanner and matches it with the ballot identification attribute for one ballot—the current Voter's—in the Ballots table of the Database.
3. The system checks the ballot-lock attribute for the ballot to ensure that the ballot may be edited.

- a. If the ballot is locked, the system disallows modification and exits the operation.
- b. If the ballot is not locked, the system performs Step 4.
4. The system queries the Database for all entries in the Votes table containing the ballot identification attribute of the ballot being modified.
5. The system generates a temporary ballot in memory from the Voter's current choices, garnered in Step 4.
6. The system displays the Voter's current choices on-screen.
7. The system performs Steps 1 – 5 in System Feature 6, "Modify choice," and returns to this feature after completion of that operation.
8. The system's operation depends on the Voter's behavior during Step 7.
 - a. If the Voter made one or more changes to the original ballot, the system performs Step 9.
 - b. If the Voter made no changes to the original ballot, the system does nothing until either the operations of System Feature 11 ("Finalize vote") are performed, or the operations of System Feature 9 are performed again.
9. The system archives the Voter's original ballot by setting its archival attribute in the Ballots table to "TRUE."
10. The system generates a new entry in the Ballots table, with a new ballot identification and bar code.
11. The system generates new entries in the Votes and Realtime_Votes tables containing the Voter's new choices. The ballot identification of each Database entry is set to the ballot identification attribute of the newly generated ballot.
12. The system performs the operations of System Feature 8, "Print ballot," and returns to this feature after completion of that operation.
13. The system decrements the tallies (by one) for the candidate(s) and/or ballot option(s) that were changes.
14. The system increments the tallies (by one) for the Voter's new choices.
15. The system performs the operations of System Feature 10, "Send real-time election results."
16. The system records an entry in its Events table indicating the success or failure of this operation and exits this sequence of steps.

3.1.9.3. Dependencies

This feature requires the successful completion of System Feature 8, "Print ballot," before it can begin execution.

This feature requires Security Features 3, "Encrypt Database tables," 7, "Encrypt outbound data," and 9, "Database login."

3.1.10. System Feature 10: Send real-time election results

3.1.10.1. Purpose of Feature

This feature allows the system to send individual votes to a central tabulation server as they are cast, before the election is ended. The full Votes table is sent when the election is declared over. This duplication of data provides anti-fraud and error-checking protection, since (as described in reference [7]) ReliaVote PE checks the Votes table it receives against the Realtime_Votes table that it has accumulated over the course of the election.

3.1.10.2. Stimulus-Response Sequence

1. The system creates a copy in RAM of the latest-added entry to the “Ballots” Database table, in plain-text. Conditional: If the latest entry is a modification of a ballot for which a paper ballot has already been printed, the system also creates a message in RAM containing the ballot identification for the Voter’s previous ballot and a request to the precinct computer to archive that ballot entry.
2. The system creates a copy in temporary memory of the latest-added entry to the “Realtime_Votes” Database table, in plain-text.
3. The system transmits a request to the precinct computer to send the real-time data.
4. The system receives a request from the precinct computer for the data.
5. The system transmits the copies of these table entries to the precinct computer.
6. The system receives an acknowledgment from the precinct computer containing the precinct computer’s success or failure to add the data to its own database. Conditional: If the system receives notification of a failure, it reverts to Step 3. Conditional: If the system receives a transmission from the precinct computer to lock itself, it immediately performs System Feature 12, “Lock voting machine.”
7. The system records an entry in its Events table indicating the success or failure of the operation.

3.1.10.3. Dependencies

This feature requires the successful completion of System Feature 7, “Cast ballot,” or System Feature 9, “Change printed ballot,” before it can begin execution.

This feature requires Security Features 3, “Encrypt Database tables,” 7, “Encrypt outbound data,” and 9, “Database login.”

3.1.11. System Feature 11: Finalize vote

3.1.11.1. Purpose of Feature

This feature locks a vote from modification.

3.1.11.2. Stimulus-Response Sequence

1. The system receives a signal from the precinct computer.
2. The system determines the signal to contain instructions to lock the ballot that was just cast.
3. The system sets the ballot-lock attribute of the last ballot in the Ballots table to "TRUE."
4. The system sends a transmission to the precinct computer that it has successfully locked the ballot.
5. The system records an entry in its Events table indicating the success of the operation.

3.1.11.3. Dependencies

This feature requires the successful completion of System Features 8, "Print ballot," and 10, "Send real-time election results," before it can begin execution.

This feature requires Security Features 2, "Verify identity of data transmitters," 7, "Encrypt outbound data," and 9, "Database login."

3.1.12. System Feature 12: Lock voting machine

3.1.12.1. Purpose of Feature

This feature allows for locking of a MyVotronic machine to block input from a voter and disable all hardware output except for the network adapter.

3.1.12.2. Stimulus-Response Sequence

1. The system receives a signal from the precinct computer.
2. The system determines the signal to contain instructions to lock the machine from accepting input from a voter.
3. The system disables all hardware input and output except from the network adapter.
4. The system sends a data transmission to the precinct computer that it has successfully locked its hardware from accepting input or producing output.
5. The system records an entry in its Events table indicating success.

3.1.12.3. Dependencies

This feature requires the successful completion of no other System Features before it can begin execution.

This feature requires Security Features 2, "Verify identity of data transmitters," and 7, "Encrypt outbound data."

3.1.13. System Feature 13: Unlock voting machine

3.1.13.1. Purpose of Feature

This feature allows SecureDRE to process a request from the precinct computer to unlock the voting machine and allow a voter to interact with the machine.

3.1.13.2. Stimulus-Response Sequence

1. The system receives a signal from the precinct computer.
2. The system determines the signal to contain instructions to unlock the machine because a new Voter is ready to vote or the election has ended.
3. The system configures the MyVotronic machine to allow voting.
4. The system sends a data transmission to the precinct computer that it has unlocked the MyVotronic.
5. The system performs System Feature 3, “Display choices to voter.”
6. The system records an entry in its Events table indicating the success or failure of the operation and exits this sequence of steps.

3.1.13.3. Dependencies

This feature requires the successful completion of System Feature 12, “Lock voting machine,” before it can begin execution.

This feature requires Security Features 2, “Verify identity of data transmitters,” and 7, “Encrypt outbound data.”

3.1.14. System Feature 14: End election

3.1.14.1. Purpose of Feature

This feature allows an Election Official to declare the election over and lock the machine from further voting for a predefined time period.

3.1.14.2. Stimulus-Response Sequence

1. The system receives one or more data transmissions from the precinct computer.
2. The system verifies that the data transmissions actually *are* from the precinct computer.
3. The system decrypts the data and determines them to be instructions to end the election and ignore all input for a given period of time.
4. The system compares its internal time to the time at which the polls are to close. Conditional: If the system’s internal time is later than the poll-closing time, it performs Step 5. If the system’s internal time is earlier than the poll-closing time, the system discards the data and logs an error in the Events database, because the presence of the erroneous data transmission indicates a problem with the precinct computer.
5. The system locks all unlocked and unarchived Ballots table entries.
6. The system performs System Feature 12, “Lock voting machine.”

7. The system records an entry in its Events table indicating the success or failure of the operation and exits this sequence of steps.

3.1.14.3. Dependencies

This feature requires the completion of no other System Features before it can begin execution.

This feature is restricted by Security Features 5, “Limit changes to Database on Election Day,” and 8, “Block all ports except two.” The feature requires Security Features 2, “Verify identity of data transmitters,” and 7, “Encrypt outbound data.”

3.1.15. System Feature 15: Send finalized election results

3.1.15.1. Purpose of Feature

This feature allows the system to transmit the final candidate tallies and individual vote Database entries to a central tabulation server.

3.1.15.2. Stimulus-Response Sequence

1. The system creates a copy in RAM of the latest-added entry to the “Ballots” Database table, in plain-text. Conditional: If the latest entry is a modification of a ballot for which a paper ballot has already been printed, the system also creates a copy in RAM (also in plain-text) of the previous entry in the “Ballots” Database table, which has been archived.
2. The system creates a copy in RAM of the latest-added entry to the “Realtime_Votes” Database table, the full “Votes” table, and the full “Tallies” table, all in plain-text.
3. The system transmits a request to the precinct computer to send the data.
4. The system receives a request from the precinct computer for the data.
5. The system transmits the copies created in Step 2 to the precinct computer.
6. The system receives an acknowledgment from the precinct computer containing the precinct computer’s success or failure to add the data to its own database. Conditional: If the system receives notification of a failure, it reverts to Step 5. Conditional: If the system receives a transmission from the precinct computer to lock itself, it immediately performs System Feature 12, “Lock voting machine.”
7. The system records an entry in its Events table indicating the success or failure of the operation.

3.1.15.3. Dependencies

This feature requires the completion of System Feature 12, “End election,” before it can begin execution.

This feature requires Security Features 3, “Encrypt Database tables,” 7, “Encrypt outbound data,” 2, “Verify identity of data transmitters,” and 9, “Database login.”

3.1.16. System Feature 16: Detect and log errors

3.1.16.1. Purpose of Feature

This feature allows the system to detect exceptions and errors and store auditable information about them in the Database for later retrieval.

3.1.16.2. Stimulus-Response Sequence

1. While performing an operation, SecureDRE encounters an error or exception.
2. SecureDRE attempts to write the contents of the system's temporary memory (hereafter the "memory dump") to the hard disk.
3. The system's behavior depends on the type of error that is encountered.
 - a. If it is a non-fatal error, SecureDRE records the system error code, identification of the machine on which the error occurred, date of the error, and time of the error in the "Events" Database table. SecureDRE also records the type of error and an error message, if provided. It then executes Step 5.
 - b. If it is a fatal error that requires that SecureDRE be rebooted, SecureDRE attempts to write the error code, date, time, error type, and error message to the hard disk. It then reboots and performs Step 4.
4. SecureDRE reboots the system. During bootup, SecureDRE reads the hard disk to determine whether the software experienced a fatal error condition the last time it shut down. Since this was the case, SecureDRE reads the memory dump from the hard disk.
5. Using the system state information in the memory dump, SecureDRE attempts to continue or restart the operation that was being performed when the error occurred.
6. SecureDRE resumes normal operation.
7. SecureDRE sends a data transmission to the precinct computer indicating that it experienced an error condition.
8. SecureDRE receives a data transmission from the precinct computer requesting the error information.
9. SecureDRE transmits a copy of the entry in the "Events" Database table to the precinct computer.
10. SecureDRE receives an acknowledgment from the precinct computer that all data were received successfully.

3.1.16.3. Dependencies

This feature requires the completion of no other System Features before it can begin execution. It does require that an error condition occur.

This feature requires Security Features 2, "Verify identity of data transmitters," 3, "Encrypt Database tables," 7, "Encrypt outbound data," and 9, "Database login."

3.1.17. System Feature 17: Clear election results

3.1.17.1. Purpose of Feature

This feature allows the system to remove all vote data and tallies from the Database after an election is over and the data are no longer needed. This operation can be performed *only* after a predetermined period of time after the day set for Election Day. It should be noted here that results cannot be deleted in part; this operation clears the entire Database table. This is an anti-fraud mechanism.

3.1.17.2. Stimulus-Response Sequence

1. The system receives one or more data transmissions from the precinct computer.
2. The system verifies the data's encryption keys.
3. The system decrypts the data and determines them to be instructions to clear the entries in the Votes, Realtime_Votes, Recount_Votes, Tallies, Recount_Tallies, and Ballots tables in its Database.
4. The system checks to ensure that enough time has elapsed and that the operation is allowed. Conditional: If not enough time has elapsed, the system displays a message indicating that the operation cannot be performed yet and exits this operation.
5. The system attempts to clear the affected tables in the Database.
6. The system records an entry in its Events table indicating the success or failure of the operation.
7. The system sends a notification to the precinct computer of its success or failure to clear the data. Conditional: If the machine failed to clear the data, the notification contains the error code and error text

3.1.17.3. Dependencies

This feature requires the completion of System Feature 14, "End election," before it can begin execution.

This feature requires Security Features 3, "Encrypt Database tables," 7, "Encrypt outbound data," 2, "Verify identity of data transmitters," and 9, "Database login."

3.2. Performance Requirements.

The features described in this section are requirements that are necessary for SecureDRE to perform in a reasonable amount of time.

3.2.1. Performance Requirement 1: Modify Database quickly

Numerous features of SecureDRE require changes to be made to the Database. Any operations involving the Database must take place in an amount of time that would be unremarkable to a member of the Voter user class. The voting machine must not appear to a Voter to slow down when performing any operation.

As is described in §3.3, subsections, numerous operations require that parts of the Database employ cryptography for data protection and machine identity verification. The cryptographic algorithms used must be executed in a reasonable amount of time and be unnoticeable to a Voter.

3.2.2. Performance Requirement 2: Print ballot quickly

Reference [4] briefly describes the hardware requirements for the MyVotronic machine, including its printer. The SecureDRE software must be able to use the printer to print a paper ballot of a Voter's electronic ballot in an amount of time that would be unremarkable to a member of the Voter user class.

3.2.3. Performance Requirement 3: Transmit and receive data quickly

SecureDRE has to receive, transmit, and process large amounts of data. This requires that the computer have access to a broadband Internet connection and a fast link between it and the precinct computer with which it will exchange data. SecureDRE itself must be able to accept the large amounts of data without slowing its processing excessively or losing any of the data.

3.3. Security Requirements.

The features described in this section are requirements that are necessary to ensure the integrity of election data generated and stored by the SecureDRE software product.

3.3.1. Security Feature 1: Limit input from keyboard

3.3.1.1. Purpose of Feature

This feature provides for the keyboard to be disabled unless a Voter has selected “Write-in Candidate” for a race.

3.3.1.2. Characteristics of Feature

As detailed in reference [4], the MyVotronic hardware contains a special keyboard. This keyboard contains the letters of the English alphabet, the space bar, the hyphen, the period, the comma, and the apostrophe.

Input from this keyboard will be discarded by the system unless a Voter is voting or modifying a vote and selects “Write-in Candidate” for a particular race. When the Voter has performed this operation, input from the keyboard will be accepted. The keyboard will be disabled again after the Voter either selects a different choice for the same race, or votes (§3.1.3).

3.3.2. Security Feature 2: Verify identity of data transmitters

3.3.2.1. Purpose of Feature

This feature restricts incoming traffic to the MyVotronic. The precinct computer (see §3.3.6) limits access to the private network, but in the event that it should fail, this security feature protects the Database from malicious operations. It requires that any data transfers processed by SecureDRE must have originated at the precinct computer or Kerberos key-management server.

3.3.2.2. Characteristics of Feature

When any data transfers are received, the system will check to ensure that they have been encrypted. The details of this system are given in reference [5], *Network Detailed Design*.

The keys will be managed by a Kerberos-compatible key management system.

If SecureDRE determines that a packet was sent by the precinct’s computer and intended for the machine that received it, then the packet will be processed. Otherwise, the packet is discarded.

3.3.3. Security Feature 3: Encrypt Database tables

3.3.3.1. Purpose of Feature

This feature provides for encryption of sensitive Database tables in the SecureDRE internal Database.

3.3.3.2. Characteristics of Feature

The Votes and Realtime_Votes tables in the Database are highly sensitive and must be protected with encryption of at least 128-bit strength when not being modified. This encryption scheme can be symmetric or asymmetric; the two options are detailed in reference [3]. The keys to this encryption system must not be known or recoverable by human users.

The entire table is encrypted rather than individual entries. This means that it is not possible to add, modify, or delete entries while the table is encrypted.

When the System initiates an operation that involves one of these tables, it decrypts the affected table, performs the operation, generates a new key, and re-encrypts the table with the new key. This means that the key is changed every time an operation is performed on one of the tables. The software generates different keys for each table.

3.3.4. Security Feature 4: Restrict data flow to Database tables

3.3.4.1. Purpose of Feature

This feature restricts traffic to sensitive tables in the SecureDRE Database.

3.3.4.2. Characteristics of Feature

The Votes and Realtime_Votes tables in the Database are highly sensitive. In addition to being protected by strong encryption and secret keys, they are protected from unauthorized modification and viewing. SecureDRE analyzes all incoming data packets to determine if they contain instructions to modify either of these tables, and if so, the packets are discarded. ReliaVote PE should not be generating such packets (refs. [6] and [7]), and their presence could indicate that the system has been compromised. Likewise, the only system that needs to request copies of the database is the precinct computer, and packets containing such requests are processed only if they originated on this computer. All packets that involve Database operations are “repackaged” by the SecureDRE software as a SecureDRE operation, so that the Database will recognize and execute them.

3.3.5. Security Feature 5: Limit changes to Database on Election Day

3.3.5.1. Purpose of Feature

This feature restricts user access to particular tables in the SecureDRE Database for a time period on and immediately after Election Day.

3.3.5.2. Characteristics of Feature

The Database contains numerous tables containing information about candidates, contests, and political parties. These tables can be modified by Election Officials using ReliaVote PE until 12:00 A.M. on Election Day. At this time all of the Database will be locked from modification (except for the Ballots, Votes, Realtime_Votes, Tallies, and Events tables) for a given period of time after Election Day ends. Only the Software user can modify these tables during this lockdown.

3.3.6. Security Feature 6: Private network

3.3.6.1. Purpose of Feature

This feature requires that all machines running SecureDRE must be part of a private network and have private IP addresses.

3.3.6.2. Characteristics of Feature

In a precinct, all MyVotronics and any other election equipment (hereafter “nodes”) will be connected to a private network with no connection to public IP addresses. The only node with direct access to public IP addresses will be the precinct computer. The precinct computer will filter traffic that is destined for private nodes. If a data transmission from the county’s computer requests that the precinct computer initiate an operation on a MyVotronic, then the precinct computer will generate a new data transmission for the intended node, with its own IP address as the source rather than the central computer’s. (It should be noted that the destination node will still discard the data packet if it contains instructions to modify a sensitive Database table.) Traffic destined for a private node originating from any other IP address is dropped.

Reference [5], *InnoVote Network Detailed Design*, contains detailed network prototypes and suggested network rules.

3.3.7. Security Feature 7: Encrypt outbound data

3.3.7.1. Purpose of Feature

This feature requires that all data transfers destined for a machine external to the MyVotronic must be encrypted.

3.3.7.2. Characteristics of Feature

In addition to being on a private network, any outbound data transmissions from a MyVotronic will be protected with encryption of at least 192-bit strength.

The feature utilizes the same cryptosystem described in Security Feature 2.

3.3.8. Security Feature 8: Block all ports except two

3.3.8.1. Purpose of Feature

This feature requires all data ports on the MyVotronic machine will be blocked from sending and receiving data transmissions except for one over which the System will exchange data with the precinct computer and another which it will use for data transfer with the Kerberos server.

3.3.8.2. Characteristics of Feature

SecureDRE will initiate connections on one data port, and this port will be used only by InnoVote software. Additionally, the client installation of Kerberos software will use a data port to communicate with the key server for the precinct network. However, a hardware firewall will be configured to disallow traffic originating from or destined for any port other than the chosen ones. The firewall must not permit any configurations that would permit incoming or outgoing traffic from ports other than these on the computer running SecureDRE. More information about this is present in *Network Detailed Design* [4].

3.3.9. Security Feature 9: Database login

3.3.9.1. Purpose of Feature

This feature requires that all accesses of the Database be made by a verified “user” that the database management system recognizes. This is to prevent unauthorized SQL querying.

3.3.9.2. Characteristics of Feature

The database management system will recognize the SecureDRE software or a SecureDRE software operation as a “user.” The software must authenticate itself when it makes any modification to the Database. The database management system will not permit anonymous SQL querying.

3.4. System Attributes.

The attributes described in this section are, unless otherwise stated, general to the SecureDRE software product rather than specific to a particular system feature.

3.4.1. Reliability

The SecureDRE software must experience normal exception-free behavior at least 99.999 percent of the time. This would correspond to no more than one exception within a 24-hour period.

3.4.2. Availability

The SecureDRE software will execute on a Direct Recording Electronic voting machine at all times. Availability is not an issue with this software product.

3.4.3. Security

The security requirements of SecureDRE are detailed in §3.3, “Security Features.”

3.4.4. Maintainability

The system must be upgradable if necessary. Any upgrades must require no changes to the existing relational schema for the Database. They must not compromise any Security Features of the software.

3.4.5. Portability

The software must execute on any MyVotronic DRE voting machine or any DRE voting machine that is functionally equivalent to the MyVotronic.