# Voting Machine Technology

December 15, 2004
Tom Trumpbour

## Introduction

In the quest to be both efficient and accurate, voting technologies have progressed since the 19th century. The overall function is to give a precise account of voter intent, thus inspiring public confidence. When this criterion is met, we accept the outcome regardless of what our vote is.

There are many hurdles to get true voter intent. Voters must have a ballot that cannot be traced back to the individual. This would prevent the purchase or some threat to gain that constituent. At the same time, auditing must be done in order to realize that the election is credible.

This short paper will delve into the history of voting machines, a study of the various technologies, a focus on Diebold design, and a summary about the DRE technology.

## History of Voting Machines

Uniform paper ballots where the first attempt to streamline the voting process. The voter makes choices in private by marking boxes on a printed form. When the voter is finished, he drops the form into a sealed box for later hand counting. The first uniform paper ballots were used in Victoria Australia in 1856. They began use in the United States in the State of New York in 1889. They account for less than 2 percent of the voting in the United States.

Mechanical lever machines are the next generation in voting machine technology. A lever closes a privacy curtain, which initializes the machine for the voter. Horizontal levers are turned for each candidate that is finalized when opening privacy curtain with lever. The counting is done with dials, which move one tenth of a rotation (36 degrees) for each digit. Its first use was in Lockport, New York in 1892. Every major US city deployed them by the 1930's. Levers counted over half of the US votes by the 1960's. Today they account for about 20 percent of US counts.

Punch cards later came on the scene in the 1960's. The voter punches holes in cards that are put in a ballot box. The punch cards are later computer tabulated at the precinct. There are two types used in the US: votematic (Candidate names correspond to numbers on the card) and datavote (Names are shown on a guide next to the chads to be punched). They were first used in Georgia in 1964. In 1996 they were used by an estimated 37 percent of the US.

Optical scan technology is much like taking a test with a number 2 pencil. The individual blackens a circle next to the candidate choice. Later, a machine reads the sheet and the votes are counted. They began use by 1980 but the exact origin is unknown. They were known to have problems in the 1980 California General Election. They were used by about 24 percent of the US by the late 1990's.

Direct Recording Electronic (DRE) is the most recent voting technology. Like lever machines, most do not currently rely on paper. The voter uses a touch screen (Diebold) or pushes buttons (Sequoia). When the vote is finalized results are stored to internal memory, disk, or a memory card. Their earliest use was in the 1970's. They grew to 8 percent use in the US by the late 1990's and today are over 25 percent.

## Study of Reliability

A group at US universities Cal Tech and MIT studied voting machine reliability. They defined a residual vote as one that no presidential preference was counted. They acknowledge that there may be some instances where such a situation is legitimate. In their averaging machines from 1988-2000, the following conclusions were made: Optically scanned, lever machines, and paper ballots had lower residual votes overall than punch cards and DRE's. They studied voting machines from 1988-2000. Here are residual numbers:

| | |
|---|---|
| Optically Scanned | 1.3 percent |
| Lever Machines | 1.4 percent |
| Paper Ballots | 1.5 percent |
| Punch Card | 2.5 percent |
| DRE | 2.7 percent |

What the study did not address are vulnerabilities outside of residual votes that could jeopardize an election.

## Diebold Voting Technology

Analysis of Diebold voting terminals has been done since the code was discovered on their public FTP site in January 2003. While Diebold officials seemed unconcerned about proprietary code being so public, they took off the material upon discovery. They have also avoided action against any Internet sites that have mirrored this code.

The code on the FTP site was for the AccuVote-ST terminal and not the GEMS back-end tallying server (the GEMS code has been leaked and on the Internet as recently as a month from this writing). The code is written in C++, which if not controlled properly, is vulnerable to buffer overflow attacks. This has not been demonstrated with this code, but with little discipline demonstrated in so many areas, it would not be a surprise.

Terminals have been configured in multiple Microsoft Windows environments, however, it is predominately used by Windows CE.

A general description of the design is as follows: A ballot is configured and stored to a file (election.edb), which can be installed by removable media (memory card or floppy) or from an outside source (modem or Internet). Control of the system is done by smart cards, including voting. The system is initialized with an administrator smart card (prompting for a PIN). Voters then vote with a smart card, which is devalued after the vote is cast. An ender or administrator smart card (the administrator requiring a PIN) is used to end the election and send the results to the GEMS management server.

There are several vulnerabilities in the system to report. One is that there was an attempt to create hardware redundancy in case there were problems by adding a removable memory drive. Because most windows systems have a separate drive letter when such a drive is present, it would be appear easy to detect this drive. Windows CE, however, mounts the drive as a directory (much like Unix only without df to double check what is happening). Merely merely taking out the memory card and creating a subdirectory with the same name could jeopardize reliability. The code only checks to see if there is a directory named "Storage Card" and no other checking.

Another problem is with smart card authentication. Smart cards are not encrypted, leaving them open to attack. Multiple voter cards can be made so that an adversary can vote that many times. But the most cleaver adversary could program a single smart card to be reused as many times as one would wish. The card type is stored in an 8-bit byte. A voter card is 1, ender card is 2, administrator card is 4, and invalidating gives the card an 8. One could program the smart card to keep the 1 value when trying to store the 8.

The smart card can also be a problem for the ender and administrator cards. Ender cards can be set up much like the voter card, with the attribute of having the 2 value instead of 1. The administrator card has what would appear to be a problem for an attacker because of a PIN. But PIN's are sent when challenged without encryption so that one could monitor the traffic to obtain it. But one doe not even need to do this. The PIN is sent in unencrypted from a stored value, which is unencrypted on the smart card itself. By merely reading the smart card, one could construct the PIN. Even without knowing a PIN, one could program one to the card and use it in the challenge process.

Ender and administrator smart cards are authenticated with a password that is on the smart card itself (the end user is not challenged here). However, the password is stored in the source, sent openly to the smart card, and unencrypted on the card too. As one can see there are multiple avenues to gain this password.

As discussed earlier, encryption is often not used or used improperly. Reporting to the GEMS management system (server) can be done over the Internet or modem. Data is sent without encryption here (which could be a problem in many wireless situations as well). A simple man-in-the-middle attack by an ISP or phone hijacker could be easily done. Totals could add up to the same sum as the intercepted traffic, if there were to be any checking. SSL/TLS could have been easily added, like many applications on the Internet.

Where encryption is attempted but done improperly is with the internal logging. Items are encrypted with the weak DES standard (not triple DES). But beyond this is that the key is hard coded into the source. Also, crypto-analysis is made easier because the initialization vector for the DesCBCEncrypt (DES in CBC mode) procedure is set to null, which defaults that variable to 0. This function works best by having a number that is as random as possible. Also, cyclic redundancy checks (CRC) are stored to another file in plain text, giving another hint for crypto-analysis.

There are many problems with the coding itself. One example is that there is no change control process to prevent malicious code from being entered into the system. Also, configuration is done in the clear within the registry. Anyone who navigates the registry can find out much about the setup. Also, the choice of C++ may be a problem, but has not been demonstrated. Java or C# are not known for buffer overflow vulnerabilities.

There are also two ways for an adversary to inflict DoS attacks. One would be to use the ender or administrator card to end the election at an opportune time. This could be accomplished at a precinct that is known to have a heavy turnout for one candidate over the other. If done at the right time, like lunch hour, the other candidate would have a great advantage.

The second DoS attack (although more speculative in nature) could be done where configuration needs to be done at the last moment and definitions are Internet or modem dependent. By bringing down a phone line or the Internet just before the election, there could be a great delay, again helping a candidate in a heavily weighed district.

This brings us to ballot definitions themselves. Votes are registered by the order of the candidates on the ballot and not the candidate name itself. If one was to change the order of the ballot definition, it could throw votes to another candidate, yet another way to sway a predominating district.

The database design has much to be desired with Diebold. They are using Microsoft Access. This is a good database for a small operation that needs a quick fix. Often developers use it as a starting point and then convert the "the real thing" like SQL Server or Oracle. As a mission critical database it lacks the robustness found in other databases like constraints, triggers, and stored procedures. There are all kinds vulnerabilities in Access, which are very easily found in any search engine. It is hard to imagine why the developers would go this route. Was it ignorance? Was it cost consciousness? Was it both? There are many more questions than answers.

Within the design of the database itself, it lacks referential integrity. An example of this is that the database does not force sequentially numbered primary keys, which makes this database an easy target to add or delete entries without any checks. One can only wonder if this was done on purpose or lack of database understanding.

## A Possible Solution

In summary, it is difficult to find an exact answer to the many problems with DRE systems. There are so many places where problems can arise. The idea of a system that would be secured yet propriety goes against what a good, secure system should be. We have been told that the most secure system stands public scrutiny. Yet opening up software like this makes it easy for a competitive company to understand what the other company is doing.

It also seems that like some other areas in computer science, solutions will take a long time and not just happen instantly (i.e. artificial intelligence). While some solutions will go a long way, there will still be something to patch.

IEEE has been publishing about a possible fix for voting technology anonymity, for example. It would involve having two encrypting vendors. Each voter could use dual keys to check their vote electronically at any time. This design would make it so that you could keep your vote anonymous since both keys are necessary. The only problem is if both venders work together, identities can be found. The challenge is to find motivations to keep either vender from working together like this.