An Analysis and Recommendations for an E-Voting System (December 2004)

Jeffrey Kler 76873017, Ivan Lau 45576022,
Aleksandar Milojkovic 78321015, Samson Zhao 41169020

Abstract – **General elections are the foundation of power for democratic governments. Recently electronic voting has been adopted by several governments for use in these wide scale elections. As with most large scale electronic systems, electronic voting may vulnerable to malicious software and tampering. For an electronic voting system to be useful it must allow for all of the benefits of information technology while operating under the legalities of standard election regulations. These election regulations are common for any democratic process and must be upheld by the technologies and protocols involved.**

Keywords— **confidentiality, cryptography, electronic voting, integrity, smartcards, software design process**

I. INTRODUCTION

SECURE electronic voting has the capability to revolutionize the way elections are conducted. Computers allow votes to be recorded and tabulated at a pace never before possible with conventional paper ballot systems. Electronic systems raise new concerns along with the potential improvements. The integrity of the election process accurately reflects the view of each individual voter. Implementing technological change in a traditional social structure such as voting, may profoundly impact the way society is governed. Social acceptance of electronic voting is essential to preserving society's faith in the democratic process. Stringent security mechanisms are needed in software, hardware and electoral protocols to maintain the validity of an electronic voting system.

Establishing a secure system for electronic voting is a key issue with security researchers and technology companies alike. As with most large scale electronic systems, electronic voting may be vulnerable to external and internal attacks. Threats such as malicious software and insider tampering must be expected within the system design and appropriate measures must be built-in to defend against them.

In this analysis, we will examine the security implementations of one such electronic voting system used in the general elections in the year 2000. Diebold in particular is a system that has been openly criticized in a report published by a group of researchers at Johns Hopkins University Information Security after they obtained a copy of source code from the "AccuVote-TS DRE" (Direct Recording Electronic) voting system. This voting system was used in a variety of counties throughout the United States, but most prominently in the states of Maryland and California. The group inspected the leaked source code and focused on flaws in Diebold's AVTSCE, or AccuVote-TS version 4 voting terminal software. Several flaws were found in the source code along with many inappropriate implementations of computer security techniques.

From the group's initial report, the manufacturer's response and independent government studies, we will attempt for further analyze the present issues with the AccuVote electronic voting system. First, we will examine the necessary requirements for a secure electronic election. Next, we will focus on the misuse of two specific technologies that could be used to implement such requirements, smartcards and cryptography. Through this analysis we will note how these security methods may be implemented and improved upon for subsequent systems.

II. GUIDELINES TO IMPLEMENTING A SECURE ELECTRONIC ELECTION

The source code of the AccuVote system has revealed security issues which may allow some of these basic outlines to be violated. The goal of electronic voting is to provide software and hardware mechanisms for a voter friendly and secure method for determining the outcome of an election. Our group believes a voting system must meet the following set of design requirements:

1.0 Scalability

Voting systems need to be able to handle very large and complex elections. Voters in North America for example are faced with different levels of elections, such as federal and municipal elections. Such systems are designed to sufficiently handle large elections.

2.0 Speed

Voting systems should produce results quickly. While this is not an absolute time limit and may vary depending on application and environment it is generally accepted that the results of a national U.S. election for president should be know to some degree of certainty within 12 hours of the closure of the last polling station. The

Manuscript received December 29, 2004. This work was initialized in part by the supported of the Department of Electrical and Computer Engineering (EECE) at the University of British Columbia under the EECE course 412: Computer Security.

majority of current electronic voting systems address this with minimal problems regarding speed.

3.0 Accuracy

The goal of any voting system is to establish the intent of each individual voter, and translate those decisions into a final tally. This involves accurately recording a vote for a candidate and keeping track of the total. To the extent that a voting system fails to do this, it is undesirable. The rules of a democratic election also state that it should be impossible to change someone else's vote, ballot stuff (vote multiple times), destroy votes, or otherwise affect the accuracy of the final tally[1]. Software security mechanisms need to be used in combination with hardware and electoral protocols to enforce these rules. There is no system that guarantees accuracy of the votes collected, and none we found with an open assurance system in place such as verifying votes.

4.0 Tamper-resistant

The voting system must also be tamper-resistant to thwart a wide range of attacks, including ballot stuffing by voters and incorrect tallying by insiders. This is the biggest security issue for Diebold in particular.

5.0 User Friendly

A voting system must be comprehensible to and usable by the entire voting population, regardless of age and disability. There have been very little complaints regarding most systems usability.

6.0 Anonymity

Secret ballots are fundamental to democracy, and voting systems must be designed to facilitate voter anonymity. In general elections in the US and Canada voters the right to remain anonymous in regards to their vote. No vote should be traceable back to the voter, once the vote has been accepted by the system. Before the voter can place a vote however, they must identify themselves to the proper electoral authorities and be verified as a potential voter. The AccuVote-TS system has issues with anonymity that when properly exploited, can reveal the identity behind each vote. This system cannot guarantee that anonymity of the votes. Each vote is recorded in a file that is eventually transferred to various networks and locations? The problem is that the votes are recorded into the file sequentially? Also recall that a poll worker tracks the order of each voter at a given terminal booth? Now if the attacker intercepted the transferred file and collaborated with the poll worker, then there would be enough information to track exactly who each voter voted for.

Despite the severity of this problem, the solution can be quite simple. The votes should be recorded on a file randomly, rather than sequentially. Previously the nth vote would be recorded in the nth line of the file. Instead, generate a random number, m, between 0 and n, and then insert the nth vote into the mth line. Avoid time stamping in order to not disclose the times that the votes were recorded.

III. Smart Cards

The 2000 US Elections employed the AccuVote TS DRE voting system. An integral part of the AccuVote-TS system was the use of SmartCards, or voter cards. The SmartCard is a piece of plastic, shaped like a credit card, with an embedded computer chip which can store digital data. On Election Day, each registered voter acquires a SmartCard from the poll workers and then submits his or her vote at a voting terminal. After the voter finishes, he or she returns the SmartCard to the poll worker who then reprograms the SmartCard for the next voter. While the use of SmartCards can be effective in improving the security of electronic voting, one shocking error in Diebold's implementation leaves the system open to many types of attacks.

The main usefulness of the user-programmable SmartCard comes from the fact that it has on chip cryptographic encryption capabilities. However, Diebold did not implement any sort of cryptographic operations into the SmartCard itself[3] which immediately forgoes any secure authentication of the SmartCards. This lack of secure authentication leaves the door open for voters to make their own fake SmartCards, also known as homebrew SmartCards. This is surprisingly easy.

SmartCards can be readily purchased online. In fact, an Integrated Circuit SmartCard, such as the type used in the US election, only costs between $7-$15 USD (java.sun.com/products /javacards/smartcards.htm).
The average voter can easily afford to purchase their own programmable SmartCard to make their homebrews. The only thing left to do is to understand the protocol used between the voting terminal and a real SmartCard. There are several ways to learn the protocol. The attacker could vote with a real SmartCard, but return a purchased SmartCard to the poll worker. At this point, the poll worker reprograms this fake card and gives it to the next person in line, the attacker's partner in crime. Now the attacker knows how a void SmartCard can be reprogrammed into a valid one. Alternatively, if the location permits, the attacker can set up a wiretap between the voting terminal and a valid SmartCard to record communication messages. This method will also give the attacker sufficient information to properly program a homebrew SmartCard.

Without a doubt, the effects of homebrew SmartCard on the voting process is devastating. Theoretically, the attacker could walk into a voting terminal with several homebrews and vote multiple times. Even worse, the attack may only use one homebrew to vote multiple times by programming the homebrew to ignore the deactivation commands from the voting terminal. Unfortunately, the Diebold system cannot differentiate between real votes and counterfeit votes made by homebrew SmartCards. This is because the Diebold system only keeps tracks of the serial numbers of the people who did not vote, and records no information about the people who did vote (although this is done with good intent; to ensure the anonymity of the voters).

There are several security issues that need to be fixed in Diebold's implementation of the SmartCard system. Our group has come up with a few solutions that may increase the security of the voting process. First and foremost, the cryptographic features of the SmartCard must be used to good effect. If each SmartCard was cryptographically encrypted, then the attacker must know the key of the encryption to have any chance of successfully programming his or her own homebrew. Inserting an unencrypted homebrew in the SmartCard reader at the voting booth would fail because authentication would not be granted since the SmartCard reader now requires proper authentication through encryption.

To increase security, the encryption on the SmartCard should be based on a hash of the voter's Social Insurance Number. The system should only accept a vote from a Social Security number if there has not already been a vote recorded from that Social Security number. This ensures that each hash is unique, and that there are never more votes than amount of eligible voters.

Now the attacker has much more work to do if he wants to make homebrews. There's no obvious solution to figure out the hash function without spending a lot of time at the voting booth trying to decrypt by brute force. Even if an attacker records how one SmartCard is reprogrammed, it does not give him any information on how he can reprogram the SmartCard for another valid Social Security number. Even simple encryption will make the task of making homebrew's far too challenging for majority of the population. For people who have a decent knowledge of encryption, there still remains the matter of extracting the hash function. To beef up security even more, double layer encryption can be used. Let h1, and h2 be hash functions, then:

$$c=h_1 (h_2 (\text{Social Security Number})) \qquad (1)$$

This method can easily be extended to multi-layer encryption systems.

The use of encryption on SmartCards acts as an authentication device to prevent the attacker from learning the protocol between the voting terminal and the SmartCard. This subsequently renders programming the homebrew almost impossible. The authentication device now built into the SmartCard is secured by multilayer encryption, which is extremely time-consuming to crack. Beyond the technical aspects, common sense and awareness should not be overlooked. The poll worker should reprogram each SmartCard in an isolated area so that people near by cannot see it. As well, the voting terminals must be checked periodically to prevent any wiretapping.

IV. Cryptography

There are instances, though, where Diebold does use encryption in an attempt to make the system more secure. However, while utilizing cryptography in a system may make a system more secure from attacks, a system that utilizes cryptography incorrectly can be detrimental as it gives a false sense of security for users. Without proper implementation, cryptographic functions will not actually accomplish any extra levels of protection.

Cryptography is used in two different places in the AccuVote TS Direct Recording Electronic (DRE) system. The first occurrence is the flash cards (PCMCIA flash cards) that store voter information as a removable hard drive on the voting terminals which utilize DES encryption. The second is for the data transfer from local election booths to regional election board that uses AES encryption.

While Diebold has been criticized for utilizing "Security through obscurity" and refusing to publish a final copy of the source code used in their electronic voting system, this stance is understandable. Security through obscurity does not imply that it is good practice to make all components known. And in such a commercial environment, as Diebold is in, one must respect the rights to intellectual property and the competitive edge it can provide.

It can also be argued that the allowing open scrutiny will motivate the programmers to write better code and allow this code to be open to a larger field of expertise. But one can also insist code must be 'bulletproof' before it is published as all potential attacks will have a blueprint to the system.

Since Diebold is utilizing a system such as DES, analysis of this particular standard is beyond the scope of this paper, it is sufficient to say that keeping any key used in their system would be of utmost importance and that minimizing the effect of a compromised key should be minimized.

"*The security of a cryptosystem must not depend on keeping secret the cryptosystem algorithm. The security depends only on keeping secret the key*"

– Auguste Kerckhoff von Niewenhof 1883

Keeping this in mind, it would not be an accepted practice to have the same key for all individual systems distributed throughout the United States. Nor would it be accepted practice to have hard coded into the software of the system. While this is exactly what was done in the implementation of the AccuVote TS. At one point the DES key is actually openly defined within the source code:

#define DESKEY ((des_key)"F2654hD4")*[3]

While this piece of source code was found in an early leak of the code to be used by Diebold, when the Maryland Department of Legislative Services conducted their independent review of Diebold this had yet to be changed. Having access to this code allows any attacker to read and program their own card as described above. As all voters' cards utilize the same code, it would be possible to bring up any number of useable ballots for the particular voter.

Mentioned in section III of this paper is the possible implementation of some chained hash functions built into the smart cards to make them more secure. This same reasoning, while absent in the analysis of the smart cards by Diebold, was applied to the flash cards.

Diebold chose to use Cipher Block Chaining (CBC) with an initialization vector to protect the integrity of the flash cars. However, in order to gain the full benefits of CBC a 'strong' set of random numbers must be chosen for an initialization vector (IV). In the early leaked software the implementation uses a NULL IV.

DesCBCEncrypt((des_c_block)tmp,*
(des_c_block)record.m_Data, totalSize,*
DESKEY, NULL, DES_ENCRYPT);[3]

Until further investigation of the final code used the benefit of the doubt will be given to Diebold and for the purpose of this paper we will assume that a sufficiently strong IV will has been chosen (clearly something other than NULL).

Similar to having the same DES key for every polling station, the same IV has also chosen for every polling station. This, it does not appear will have changed in any final release of the product. (Remember that both are hard coded into the software). These can clearly be improved upon by having a unique code for each machine, or at least each County that is not hard coded into all machines. This would make certain all terminals are not susceptible to the same attack from the same leaked passwords and/or IV's.

There is a glaring mistake in this analysis though. This default DES code of 'F2654hD4' can be used in conjunction with a PIN to reset, by election officials, the code on each card and terminal from the default password used to secure the voter cards and voting terminals. While it is unclear exactly how many voting stations actually did take advantage of this to correct flaw of having the same code for every machine, any number of reasons exists as to why this feature would not be taken advantage of: lack of proper training, lack of awareness of a threat – both examples of lack of physiological acceptance.

But is this actual security of the system, or just a case of a false sense of security. It was actually found that this programmed PIN is actually stored in virtual clear text on the code. Not encrypted or digitally signed. i.e. if you wanted to know this new 'secure' password all you need to do is read the card using the default DES password. This PIN number is also stored, again with little protection, on the PCMCIA flash cards in the voting terminal.

Additionally improvements may also include message authentication. Some form of redundancy check or checksum implemented and would be well within the capabilities of the smart cards and hardware of the voting terminals. Common practice would be to first encrypt the data to be stored and then compute a checksum (such as HMAC-SHA1) of the cyphertext. This could then detect any tamping with the plaintext on the flash cards.

V. Assurances and Auditing

The voting system provided by Diebold in the 2000 US election uses Direct Recording Electronic (DRE) terminals. DRE terminals are convenient for voters because they can make their selections easily on a touch screen display. Touch screen voting removes many of the cumbersome and often confuSocial Security numberg paper ballot selection methods. A list of candidates is displayed to the voter and after receiving the voter's input, the decision is stored on computer memory and the appropriate counters are updated. These results are recorded and saved on each individual voting terminal with the counts to be later sent to Diebold's GEMS back-end election management system for final tabulation. The voter is given verification of their choice by a final confirmation screen at which they have the option to make any last changes to their vote preference. There is no physical receipt given to the voter which would indicate their selection. This places a high level of responsibility on the electronic election system for managing and protecting the integrity of the data contained within.

To increase voter confidence in electronic voting, it is recommended that paper verification also be dispensed after every vote. The voter should have physical proof that their vote was entrusted to the system. Allowing the voter to confirm their selection through this means would be a beneficial step to adding credibility to a system such as Diebold's AccuVote TS system. Government legislators agree with such paper trail features, for instance, California Secretary of State Kevin Shelly announced that "[b]eginning July 1, 2005, no county or city may purchase a touch screen voting system that does not include an accessible voter verified paper audit trail (VVPAT)"(p.30, News Release). VVPAT is a method of providing assurance certificates to voters. It gives proof that their vote counted and it was recorded to the correct candidate.

From the leaked version of the AccuVote-TS DRE source code, the team from Johns Hopkins University also noted in their paper that voting records and audit logs stored on the AccuVote-TS DRE terminals were potentially vulnerable to modification. The records and logs were encrypted with simple DES cryptography and used the same key in the all versions of the source code. The hardcoding of keys into a program's source code is a poor approach to encryption (see the section on cryptography above). If the same compiled program image is used on every voting terminal, an attacker with access to the source code could learn the key and thus modify voting and auditing records on any machine (*Kohno et al. "Analysis of an Electronic voting system*"). A stronger version of encryption such as AES and a random key generator should be used in such an implementation.

Also revealed in the source code was a potential security risk regarding how the data was transferred from the voting terminals to the GEM back-end central server. From their analysis of the source code, it appears to be the case in at least some areas; no cryptography is used when they should be. Voting data is to be sent in cleartext. Some functions within the source could were programmed to write votes in cleartext to a socket connection. There was no cryptography on that socket and no checksum attached to the message to prove that the data arrived as it was originally sent. An adversary could modify the data in transit and the election officials would have no evidence that the data was corrupted.

Cryptography should always be used when sending sensitive data over communication mediums. A Secure Socket Layer (SSL) connection supporting MD5 or SHA would be an improved method of sending data to the back-end server.

VI. Putting the Whole System Together

In bits and pieces we have analyzed an electronic voting system and suggested how an improved system may have approached for that piece. We will now put these pieces together to show how an entire working system could be set up for improved security and assurance.

Starting with the SmartCards that the voters receive from the Election Official. These SmartCards will be encrypted with a unique hash based on the Social Security number of the voter being issued the card, and time stamped and encrypted with a unique key for that polling station. The time stamp ensures the SmartCard cannot be copied and used again later. The Social Security number hash for every eligible person is precomputed for that county and stored on those voting terminals. This is based on the voters list available for election officials anyway.

The SmartCard then verifies itself to the voting terminal. This is done by using the hash of Social Security number. This hash is check against a log of hashes of people that have already voted and eligible voters. If this hash appears in the log of people that have already voted then the machine know the person attempting to vote has already done so. Or if doesn't appear in the list of eligible voters the terminal knows they should not be eligible. It also checks the time of verification to ensure that it is not past a certain time window.

Once the voter has placed their vote the voting terminal records, in a database, the hash of the Social Security number of the person who has just voted. Every entry into the database is randomized so that no chronological order can be recovered. This hashed value is shared with all other terminals. This ensures that a person can only vote a single time while still remaining anonymous. If they attempt to vote again, the same hash will be produced from their Social Security number (hash is not based on time of verification) and will already be in the log and they will be unable to vote again. This shared hash value between the local polling station network will be encrypted with some HMAC or SHA1 so that any tamping with the text while in transit would be detected.

When a person is finished voting the terminal will issue a receipt of who they voted for as well as perform a 'back off' wait period of a short period to further prevent ballot stuffing by voting multiple times.

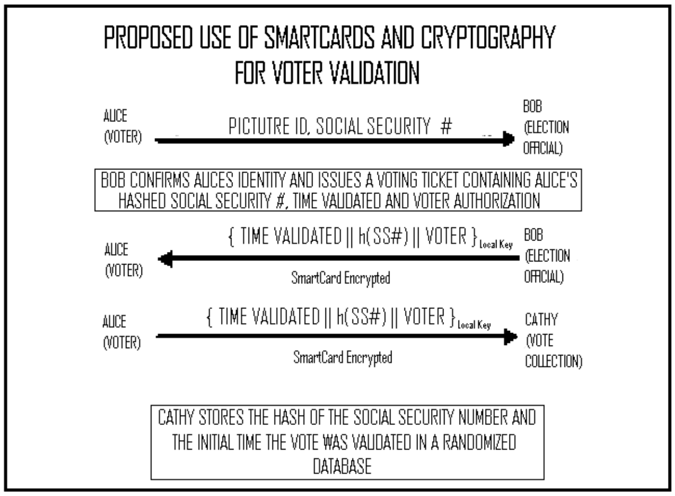
PROPOSED USE OF SMARTCARDS AND CRYPTOGRAPHY
FOR VOTER VALIDATION
ALICE (VOTER)
PICTUTRE ID, SOCIAL SECURITY #
BOB (ELECTION OFFICIAL)
BOB CONFIRMS ALICES IDENTITY AND ISSUES A VOTING TICKET CONTAINING ALICE'S HASHED SOCIAL SECURITY #, TIME VALIDATED AND VOTER AUTHORIZATION
ALICE (VOTER)
{ TIME VALIDATED || h(SS#) || VOTER } Local Key
SmartCard Encrypted
BOB (ELECTION OFFICIAL)
ALICE (VOTER)
{ TIME VALIDATED || h(SS#) || VOTER } Local Key
SmartCard Encrypted
CATHY (VOTE COLLECTION)
CATHY STORES THE HASH OF THE SOCIAL SECURITY NUMBER AND THE INITIAL TIME THE VOTE WAS VALIDATED IN A RANDOMIZED DATABASE

Fig 1. Key Management Diagram

VII. Conclusion

So far we have analyzed violation of confidentiality; DES passwords and PIN numbers, violation of data integrity; and 'ballot stuffing' with extra votes from spoofed voter cards. By taking the steps mentioned, prevention of repudiation of source on the smart cards can be archived using cryptography; detection of tampering can be collected using checksums on flash cards. Although the perfect system is still very difficult to implement, government regulations and requirements can help improve security. While more government bodies are adopting technologies to increase work efficiency, electronic voting system must need a rapid evolution to keep up with the pace. Instead of banning such machines, people should take a step forward to make the systems better so that the world get move forward.

References

[1] Bruce Schneier "A Weblog Covering Security and Security Technology" Published Nvember 10th 2004, Available: http://www.schneier.com/blog/archives/2004/11/the_problem_wit.html

[2] Definition and overview of electronic voting defined at: http://en.wikipedia.org/wiki/Electronic_voting#Overview

[3] Avi Rubin "An Analysis of and Electronic Voting System," *IEEE Symposium on Security and Privacy, May 2004 Available: http://www.avirubin.com/vote.pdf*

[4] *Identity Based Encryption (IBE) on Smart Cards Available: http://www.inside.com/credentialing /article.php.3430921*

[5] Diebold Election Systems official response to Paper Published by Hopkins Avi Rubin in IEE, "Checks and balances in elections equipment and procedures prevent alleged fraud scenarios'" Available: http://www2.diebold.com/checksandbalances.pdf