



# Security Assessment of the Diebold Optical Scan Voting Terminal

A. Kiayias      L. Michel      A. Russell      A. A. Shvartsman

UConn VoTeR Center and  
Department of Computer Science and Engineering,  
University of Connecticut  
{akiayias,ldm,acr,aas}@cse.uconn.edu

with the assistance of  
M. Korman, A. See, N. Shashidhar, D. Walluck

October 30, 2006

## Abstract

We present an independent security evaluation of the AccuVote Optical Scan voting terminal (AV-OS). We identify a number of new vulnerabilities of this system which, if exploited maliciously, can invalidate the results of an election process utilizing the terminal. Furthermore, based on our findings an AV-OS can be compromised with off-the-shelf equipment in a matter of minutes even if the machine has its removable memory card sealed in place. The basic attack can be applied to effect a variety of results, including entirely neutralizing one candidate so that their votes are not counted, swapping the votes of two candidates, or biasing the results by shifting some votes from one candidate to another. Such vote tabulation corruptions can lay dormant until the election day, thus avoiding detection through pre-election tests.

Based on these findings, we describe new safe-use recommendations for the AV-OS terminal. Specifically, we recommend installation of tamper-resistant seals for (i) removable memory cards, (ii) serial port, (iii) telephone jacks, as well as (iv) screws that allow access into the terminal's interior; failure to seal *any single one* of these components renders the terminal susceptible to the attack outlined above. An alternative is to seal the entire Optical Scan system (sans ballot box) into a tamper-resistant container at all times other than preparation for election and deployment in an election. An unbroken chain of custody must be enforced at all times. Post-election audits are also strongly advised.

*The Diebold AccuVote Optical Scan voting terminals described in this report are going to be used in November 2006 election in several precincts in the State of Connecticut. The terminals are provided by the LHS Associates of Massachusetts. VoTeR Center personnel assisted the Office of the Connecticut Secretary of the State in developing safe use procedures for the Optical Scan terminals for this election. The procedures in place for the election includes strict physical custody policy, tamper-resistant protection of the equipment, and random post-election audits.*

## 1 Introduction

The subject of this paper is the **AccuVote Optical Scan** voting terminal (AV-OS) manufactured by Diebold, Incorporated, Election Systems division.



Figure 1: The AccuVote Optical Scan voting terminal (AV-OS). The terminal is shown prior to it being locked to the ballot box, with its front panel visible and showing two control buttons (lower left corner) and the memory card slot with the card sealed in (lower right corner).

An important benefit of using the optical scan technology in electronic voting systems is that it naturally yields a voter-verified paper trail—the actual “bubble sheet” ballots marked by the voters. This differentiates optical scan electronic voting from DRE (direct recording electronic) electronic voting terminals (such as the Diebold AccuVote TS and TSx terminals) that provide a digital interface for voting during the elections. We note that the current generation of the DRE terminals—especially paperless ones—have received substantial criticism due to a number of critical security vulnerabilities, such as those reported in [1, 2, 7]. Even when a DRE terminal is equipped with a printer, the computer-generated paper trail cannot be directly considered voter-verified, and it is possible for a faulty DRE to print spontaneous ballots while unobserved. Further development of the DRE technology is necessary for it to become a trustworthy alternative.

While optical scan voting is freed from some of the perils of paperless trails or computer generated paper trails, the election still relies on the terminal to electronically add the votes and report the results; this introduces the possibility of attacks that interfere with these basic tabulation and reporting tasks. Such an attack against the AV-OS was demonstrated recently in [3]. This attack was particularly devastating as it initialized the counters of the terminal to negative or positive vote counts while still forcing the machine to report a valid zero-count initialization. This can lead to biased election results and corrupted election counts. The operation of the AV-OS system is in part governed by the instructions stored in a *memory card* that is inserted into the terminal for the duration of the election. The attack of [3] employed a memory card reader/writer to modify the card prior to election and bring it to an *invalid initial state*. When a maliciously altered card is used in an election, it records biased results that are successfully tabulated by the terminal.

Given that the attack in [3] required tampering with the memory card directly, one way to mitigate the attack is to somehow ensure that the memory card stays in place sealed into the terminal throughout the period that the machine is in use or is in transit to and from the polling places. Alternatively (and most effectively) one could employ a cryptographic integrity check, however this would require modifications to the firmware

of the system (presumably by the manufacturer). A second way to mitigate the attack would be to execute a pre-election test, hand-count the ballots, and compare this to the report of the terminal.

Given the facts summarized above, the pressing question is whether the security measures of (1) sealing the memory card into the terminal, and (2) performing pre-election testing with hand-counted ballots, are sufficient to prevent an attack against an election employing the AV-OS.

Our findings answer this question in the negative.

In particular we show that *even if the memory card is sealed and pre-election testing is performed*, one can carry out a devastating array of attacks against an election *using only off-the-shelf equipment and without having ever to access the card physically or opening the AV-OS system box*. Our attacks include the following:

1. **Neutralizing candidates.** The votes cast for a candidate are not recorded.
2. **Swapping candidates.** The votes cast for two candidates are swapped.
3. **Biased Reporting.** The votes are counted correctly by the terminal, but they are reported incorrectly using conditionally-triggered biases.

Our attacks exploit the serial communication capability of the AV-OS and demonstrate how the attacker can easily take control of the machine and force it to compromise its sealed-in resident memory card. Moreover, we demonstrate how one can make the AV-OS appear to be uncompromised to an evaluator that performs a pre-election test by voting hand-counted ballots, or to an evaluator that examines the audit reports that are produced by the terminal. A corrupted terminal will in fact appear to be faithfully reporting any election procedure that is conducted prior to the day of the election, only to misreport its results on the day of the election.

We also present a low-tech “digital ballot stuffing” attack that is made possible due to the mechanical characteristics of the optical scan reader. This simple attack enables any voter to vote an arbitrary number of times using two Post-it® notes. This attack makes it imperative to have the terminal under constant supervision during elections.

The vulnerability assessment provided in this paper is based only on experimentation with the system. At no point in time had we used, or had access to, internal documentation from the manufacturer or the vendor, including internal machine specifications, source code of the machine’s operating system, layout of the data on the memory card, or the GEMS ballot design and tabulation software. We developed attacks and software that compromises the elections from first principles, by observing system’s behavior and interaction with its environment. Based on this fact, we conclude that attackers with access to the components of the AV-OS system can reverse-engineer it in ways that critically compromise its security, discover the vulnerabilities presented herein and develop the attacks that exploit them.

## 2 Basic Characteristics of the System

Our findings are based on the evaluation of an AV-OS system that was delivered to us by LHS Associates of Methuen, MA as a part of an evaluation on behalf of the State of Connecticut. The AV-OS election system consists of two components: the AccuVote Optical Scan voting terminal (the AV-OS terminal) and the ballot design and central tabulation system (GEMS, for Global Election Management System). These components have the following characteristics:

- The GEMS software is installed on a conventional laptop PC and includes a ballot design system and a tabulation system.
- The specifications of an election are downloaded onto a 40-pin 128KB Epson memory card present in the AV-OS. It should be noted that the memory card has been discontinued by Epson, and no reader/writer for this type of medium is readily available in the market.
- The AV-OS systems provided to us contained the firmware version 1.96.6. It is equipped with an optical scanner, a paper-tape dot-matrix printer, a LCD display, a serial communication port, and telephone jacks leading to a built-in modem. For election deployment the system is secured within a ballot box so that no sensitive controls or connectors are exposed to the voter.

### 3 Security Vulnerabilities

We briefly describe the new vulnerabilities that were discovered during our evaluation process. A detailed presentation of these vulnerabilities is available in an extended version of the report that can be provided on a need-to-know basis.

**The AV-OS leaks the memory card contents:** The AV-OS terminal allows any operator to obtain a dump of its installed memory card contents without any authentication control. In particular, given access to an AV-OS machine one can obtain all the information that is stored in the memory card in a matter of seconds. In order to obtain this information, it is sufficient to use an off-the-shelf RS-232 serial cable (null modem cable) and a laptop. The AV-OS performs no authentication test to ensure that a trusted system is present on the other side while the dump is delivered in cleartext form. Moreover, the terminal does not prompt the operator for a password in order to produce such memory dump. It is easy to identify the election data when observing a memory dump; other sensitive information, including the *password (PIN) and audit records* associated with the memory card can also be reconstructed from the dump. Alternatively, the same dump can be obtained by using the built-in modem on the AV-OS to transmit the data to a remote PC.

**The communication between AV-OS and GEMS is unauthenticated:** During the initialization of a machine for election the GEMS system communicates with the AV-OS terminal to write the initial election setup to the memory card. No encryption or cryptographic authentication is performed during this transmission. The serial line protocol does use a cyclic redundancy check (CRC) mechanism for error control. While the CRC polynomial used is standard, the details of the protocol are undocumented by the manufacturer; as such, this is a de facto lightweight authentication mechanism. However, it is possible to reverse-engineer the whole protocol, including the CRC scheme formula (as we have done in our assessment). The lack of cryptographic authentication opens the possibility for an unauthorized attacker computer to impersonate the GEMS system to the terminal (this is one of the ingredients of our main election compromising attack in the next section).

**Executable code within the AV-OS memory card:** Each memory card contains executable code that is used for printing the reports. The code is written in a proprietary symbolic language. Such executable files are identified as .abo (AccuBasic Object) bytecode. The possibility to modify the code that prints the results opens the possibility to corrupt machines and coerce them into misinterpreting their counters. The presence of conditionals and arithmetic in the language enables bytecode “malware” to operate even conditionally on the state of the machine and thus appear to operate properly in some occasions

while misreporting the results in others. While this vulnerability was already known from [3, 7] it was not employed as a tool to conditionally misrepresent the outcome of an election (but rather as a tool to hide a corrupted initial election setup).

**Multiple feeding:** The sensor that detects that a ballot has been fully inserted into the optical scan slot is positioned in the right hand side of the feeder. This opens the possibility for multiple feeding if voters are left unattended during the time that they insert their ballot into the terminal (cf. §4.2).

## 4 The Attacks

We now present new attacks against the AV-OS system that use the vulnerabilities described above. The first attack entirely compromises the election process assuming that the attacker has a few minutes of access to the AV-OS terminal prior to election time. The second attack shows how voters can vote multiple times using the same ballot if they are left unattended to use the terminal during an election.

### 4.1 Compromising the Election

By compromising the election we refer to an attacker's capability to put the AV-OS in a state where it miscounts the ballots that are inserted into the machine. For example an election would be compromised if the votes received by two candidates are swapped or if the votes of a candidate were nullified.

To streamline our attack, we have developed a proof-of-concept software package that processes card dump data, extracting the ballot layout, password (PIN), and audit information, and computes a serial payload to reprogram the card. We emphasize that our software was developed by observing the AV-OS system during normal operation, without access to any technical information about the system, its internals, or access to the source code of AV-OS or GEMS. Specifically, the attack was developed with precisely the same information and access to the system that is normally available to, for example, election administrators (poll workers and other town officials).

Equipped with a laptop and a regular RS-232 null modem serial cable, an attacker needs only to gain physical access to an AV-OS terminal prior to the election. Furthermore, the attacker needs no knowledge of the particulars of the election he is to undermine (such as exact candidates' names, ballot layout, precinct names, or any kind of passwords). The whole process can be completed in a matter of a few minutes. In the following we perform a step-by-step demonstration of the attack.

**Step 1 : Gaining physical access<sup>1</sup>.** In Figure 2 the AV-OS terminal is shown locked within the ballot box. This would (presumably) be the state that the terminal is found prior to the election. At this stage, the system has been initialized with all the election data and its removable memory card is sealed with a tamper evident seal.

The first thing an attacker must do is gain access to the front side of the AV-OS that is concealed by the ballot-box. If the box is unlocked or the attacker has the keys this is straightforward. We note that the locks used are regular pin tumbler locks similar to those found in filing cabinets, office drawers or other standard computerized equipment. If the attacker lacks the key, picking the lock can be done in a short amount of time ranging from seconds to minutes (it is feasible even for someone who has never done it before using information available online, e.g., [6]). Picking the lock requires no special equipment: in fact two standard paper clips are sufficient, see Figure 3.

---

<sup>1</sup>If the attacker has access to the election-ready voting terminal prior to its being locked within the ballot box, proceed to Step 2.



Figure 2: The AV-OS terminal installed within the ballot box and locked.

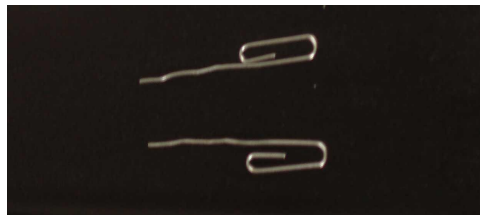


Figure 3: The two clips an attacker can use to pick the AV-OS ballot box front lock and gain access to the machine.

Once the lock is opened, the front side of the machine is freed and the attacker can access the Yes/No buttons located on the left of the front panel of the AV-OS, see Figure 4.

**Step 2 : Dumping the memory card contents.** Once the AV-OS terminal is accessible from the front, the attacker can pull it slightly outwards and obtain access to its back side. There, a number of standard connection ports are available including a RS-232 serial port and a telephone line jack. The attacker uses a standard serial cable to connect the machine to her laptop, see Figure 5.

In order to prepare for the attack the laptop must capture the data sent to its serial port by the AV-OS. Though we have written software which automates this portion of the attack, a standard terminal emulator would suffice.

Once the serial cable is in place, the attacker turns on the machine using the on/off switch located on the right of the machine's back panel while simultaneously depressing the two buttons on the front panel. This results in the AV-OS entering in a special diagnostic mode. The terminal asks for no password or other identification from the operator in order to enter into such a mode. One of the options that is available to the attacker in this mode is to dump the contents of the installed memory card through the serial line. This is the option that the attacker selects and the AV-OS dumps the card contents, see Figure 5.

It takes roughly two minutes to receive (and parse) the card dump that the AV-OS transmits. The dump of the card is sent in cleartext and the only component that is hidden is the PIN that enables the attacker to



Figure 4: The front side of the AV-OS with the Yes/No buttons accessible.

enter into a special “supervisor mode.” This is the mode that poll-workers have access to during election time. The 4-digit PIN is contained in an obfuscated form at a fixed location in the dump. As part of assessment, we reverse-engineered the method used to obfuscate the PIN. The election compromising software de-obfuscates and prints the supervisor mode access PIN, see Figure 6. In addition, our software decodes the “audit history” that appears in the card dump, including the entry containing the initialization timestamp for the card as well as any other entry in the transaction log of the terminal.

In the same screenshot our main menu is presented that has the following options: (1) neutralize candidate votes, (2) swap candidate votes, (C) print candidate list, (D) display election info, (Q) quit and send data. Using the options (C) and (D) the attacker can obtain all information about the election including the ballot layout.

**Step 3 : Ballot design remapping.** In order to understand the specifics of the attack, we overview the election setup of the AV-OS system. Each candidate and race has a unique identifier. The candidates for each race are encoded together with an  $(x,y)$  coordinate (cf. Figure 6), which corresponds to the bubble on the paper ballot sheet that the voters mark in order to vote for that particular candidate. The ballots are printed taking into account this configuration. The correct correspondence between printed ballots and internals of the memory card is essential for the election to go through uncompromised. This correspondence is one of the aspects of the election system that the attack subverts.

To neutralize a candidate in a specific race, the attacker simply maps the  $(x,y)$  coordinate of the candidate to some location that is beyond the ones used for the election (note that most coordinates are in fact unused; thus it is trivial for the software to recover such a position). In the current implementation of the election compromising software the location selected for neutralizing a candidate whose coordinates are  $(x,y)$  is  $(x - 1, y + 1)$ . The  $(x - 1, y + 1)$  pair is suitable as this choice will not affect the value of the *checksum* that the terminal computes from the ballot layout data.

An equally devastating attack is swapping two candidate’s votes. Following the previous rationale if the bubble coordinates assigned to candidate *A* are  $(x_A, y_A)$  and the bubble coordinates assigned to candidate *B* are  $(x_B, y_B)$ , by simply swapping the coordinates one effectively makes AV-OS count a vote for candidate *A*





Figure 5: (*Left*): The setup for compromising the AV-OS. A standard laptop is connected through its serial port to the serial port of the AV-OS machine. (*Right*): Dumping the memory contents through the serial line. Anyone with physical access to the AV-OS can perform this operation since this function does not require any authentication. The number shown in the LCD screen is the amount of remaining bytes before the dump is completed (each card holds 128Kb of data).

as a vote for candidate  $B$  and vice versa.

These modifications are built-in into the election compromising software that also includes additional payloads for biasing the reporting functionality of the terminal. What needs to be performed next by the attacker is to use the AV-OS to reprogram the memory card with this altered election data.

**Step 4: Adjusting the AV-OS clock to agree with the card's initialization timestamp.** When the election compromising software processes the dump of the memory card, it also recovers the time and date at which the card was originally programmed for the election. To insure that this timestamp is preserved in the audit history of the new image of the card to be created in Step 6 below, the attacker would need to reset the clock of the AV-OS so that it agrees with the recovered timestamp. The option to (re)set the clock appears in Diagnostic Mode, obtained by restarting the machine with both buttons pressed.

**Step 5: Temporarily disabling the AV-OS printer.** When the AV-OS terminal is initialized it prints a tag that can be used for auditing the system and contains the date and time of the initialization as well as some other control information. Given that the attacker will reinitialize the system, in order to prevent the AV-OS from printing such tag, the attacker must disable the printing functionality by selecting the corresponding choice available in the “supervisor menu” of the terminal that is accessible by using the de-obfuscated PIN. This step is optional as the attacker may simply discard the printout, nevertheless the fact that the attacker can disable



```

File Edit View Terminal Tabs Help
Terminal Terminal Terminal Terminal
PIN:7251
Location:WESTPORT, CONN.
Election:MUNICIPAL ELECTION
Options:
(1) Neutralize candidate votes
(2) Swap candidates votes
(C) Print candidate list
(D) Display election info
(Q) Quit and send data
Choice:c

          name      Bubble position(x,y)
BOARD OF FINANCE:
  R.GAVIN [REDACTED] (21,10)
    THOMAS C [REDACTED] (21,13)
      RALPH [REDACTED] (21,16)
    CHARLES [REDACTED] (21,19)
      STEVEN [REDACTED] (18,10)
    KEVIN A [REDACTED] (18,13)
BOARD OF EDUCATION:
  EDWARD M [REDACTED] (21,22)
    LEWIS [REDACTED] (21,25)
      MARK H [REDACTED] (18,22)
      MARY R [REDACTED] (18,25)
      STEPHEN M [REDACTED] (12,22)
      ROBERT HALE [REDACTED] (12,25)
      ROBERT M [REDACTED] (12,28)
BD OF ASSESSMENT APPEALS:

```

Figure 6: (*Top part*) : The main menu of the election compromising software. The de-obfuscated PIN is prominently presented. (*Bottom part*) : The listing of candidates for some of the races and the corresponding bubble sheet coordinates for each candidate. (This snapshot is touched-up to black out the last names used in this fictitious race.)

the printer makes the attack more stealthy.

**Step 6 : Impersonating the GEMS system.** Once the AV-OS clock is reset and the printer is shut-off the attacker sets the AV-OS terminal in supervisor mode.

In supervisor mode, AV-OS can format the contents of the memory card and accept communication from the GEMS system to initialize the election. The attacker takes advantage of the fact that the AV-OS does not use any strong cryptographic identification check to authenticate the sending entity and hence it can impersonate the GEMS system.

Using the election compromising software the attacker prepares a forged election payload. The preparation of this payload is based on the reverse engineering of the communication between the GEMS system and AV-OS that we performed as part of our vulnerability assessment. The software prepares a fake communication transcript that appears to be originating from GEMS. The transcript contains the election details recovered from the memory dump together with a number of malicious alterations such as candidate swaps, candidate neutralizations and corrupted bytecode reporting functionality.

In this particular run of the attack we made the following choices (see Figure 6):

- The votes of “Thomas C” and “Kevin A” in the Board of Finance race are swapped.
- The candidate “Mark H” in the Board of Education race is neutralized.



Figure 7: (*Left*) : The attacker enters the PIN to enter supervisor mode. (*Right*) : The AV-OS is requesting communication from the GEMS system to overwrite the memory card contents with the forged election setup.

In Figure 7 we show how the attacker enters the 4-digit PIN that was recovered from the memory dump to gain access to the options of the supervisor mode of the terminal. In order to start the machine in supervisor mode the unit needs to be turned off and restarted while simultaneously depressing the ‘Yes’ button. Subsequently the attacker chooses to erase the memory card contents, and the card is formatted. Once the contents of the memory card are erased the unit would request to be initialized from the GEMS system. In Figure 7 the AV-OS terminal requests communication from the GEMS system. The attacker furnishes to the terminal the forged communication transcript.

**Step 7 : Completing the attack.** Once the forged communication is transmitted through the serial port the compromise of the terminal has been successfully completed. The attacker will reset the clock to the current time using the diagnostic mode and will activate the printer.

After this step, the AV-OS terminal will be found by poll-workers in its expected pre-election state. The terminal will appear to be functioning normally for all operations during the election. Interestingly, the terminal is not even in an invalid initial state after a card has been compromised in the way described above. The defect is found only in the mapping between the candidates and the bubbles in the printed ballot sheets that has been rearranged in malicious intent.

The total time required to compromise the card is only a few minutes, depending on the dexterity of the attacker in picking the lock of the ballot box. If the attacker possesses the key or minimum lock-picking expertise the locks can be picked in about 30 seconds and the whole attack can be carried out at a leisurely pace in less than 5 minutes.

#### 4.1.1 Test Election

We have conducted an election with the compromised AV-OS terminal to illustrate the attack’s effectiveness. We have prepared 10 ballots as shown in Figure 11 in the appendix where we selected 7 votes for candidate “Thomas C” and 5 votes for candidate “Kevin A” in the race for the “Board of Finance.” We also voted thrice for candidate “Mark H.” in the race for the “Board of Education.” As shown in the report presented in Figure 12, the results of candidates “Kevin A.” and “Thomas C” are reversed with “Kevin A” receiving 7 votes and “Thomas C” receiving 5 votes. On the other hand the “Mark H.” appears to have received no votes at all.

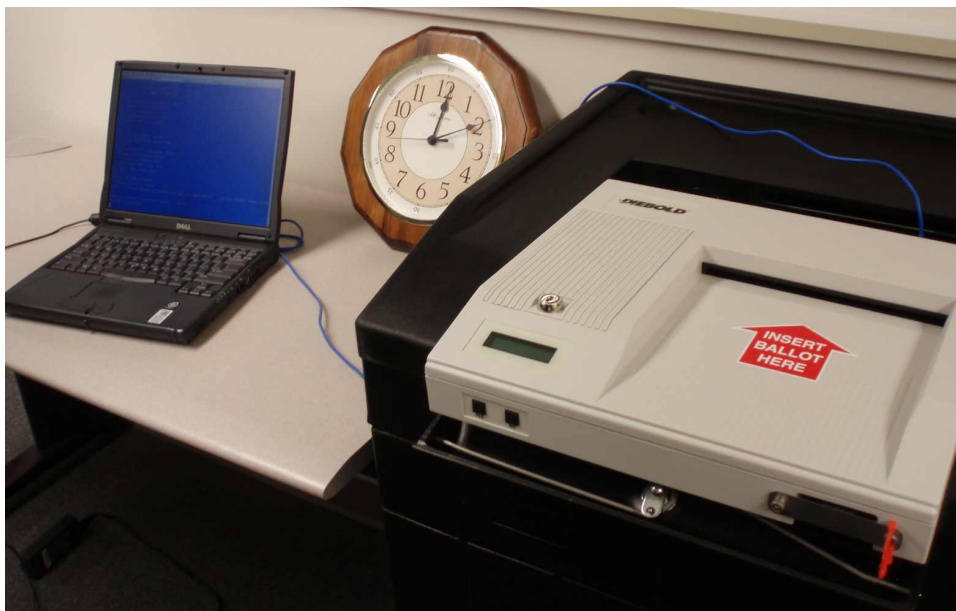


Figure 8: The AV-OS terminal has been compromised. The election specifications that are stored in the memory card contain an incorrect mapping of candidates to bubble locations, replacing the mapping necessary to conduct a proper election as configured on the printed ballots.

#### 4.1.2 Identifying a Compromised Machine

The election compromising software takes special measures to conceal the tampering of the terminal and make the detection of a compromised machine difficult even when the election officials follow the recommended procedures to test the integrity of the system. In particular, the election compromising software embeds the following functionality into a corrupted machine: a compromised AV-OS terminal records the actual results improperly but *when the poll-workers execute a test election* prior to opening the polls it still prints out the results correctly.

This double deceitfulness of a compromised machine — to behave improperly in the real election but behave properly when tested — can be achieved as follows: the report functionality of the terminal is altered by the election compromising software so that it corrects its misaligned counters in the event that the ballot count is too low (which would correspond to the case when the poll officials test a small batch of hand-counted votes) or when the date and time is prior to the real election time.

In other words, in standard computer security terminology, the attacker can plant a “time-bomb” in the terminal. Before the election, the program in the terminal’s card inverts the swapped counters to conceal the malicious behavior (the swapping of votes). When the time of the election comes, the illicit behavior is triggered automatically. This sensitivity to time will prevent poll-workers that perform the standard test procedures from revealing that a machine is compromised prior to the election.

#### 4.1.3 Compromised Election Results

An election is deemed corrupted when the miscounted results get tabulated into the overall election totals. If this is performed manually using the printed receipts that are produced by a corrupted terminal, the corruption

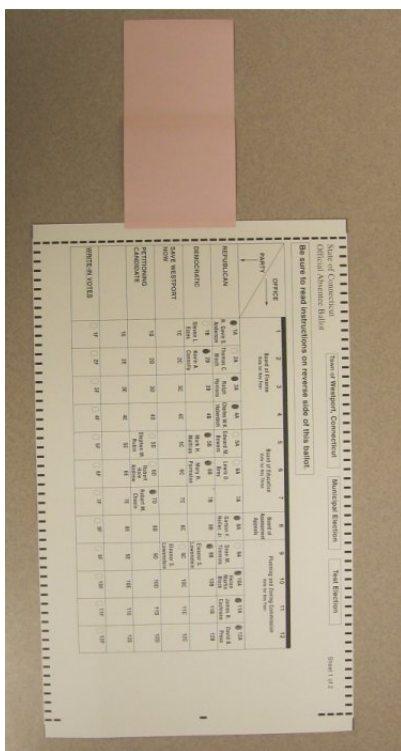


Figure 9: The prepared ballot used for the re-voting attack. Ballot stuffing is as easy as obtaining a couple of standard Post-It notes if the terminal is not closely monitored during an election.

of an election would be immediate. The results can also be tabulated electronically, by consolidating memory cards using a terminal and communicating such results to the central tabulation system implemented in GEMS. The compromised cards that contained the improperly aligned counters are accepted by the central tabulation system without any warning or any other indication that they may be corrupted.

## 4.2 Multiple Voting Using Two Post-It® Notes

In this section we present a simple low-tech attack that is based on the following facts regarding the ballot feeding mechanism of the AV-OS terminal:

- The ballot-feed sensor is located on the right side of the slot. Feeding paper into the left side does not trigger the feed mechanism.
- Once a ballot is fed into the AV-OS, the rollers cease. It is thus possible to retract a ballot from the other side of the rollers. This is easily done even when the AV-OS has been properly locked into position atop the ballot box. Moreover, this can be done very quickly, so that the amount of extra votes is only limited to the amount of time the voter is able to spend alone with the ballot box on election day.
- The machine is unable to recognize ballots that have already been cast. Although the AV-OS verifies an election identifier which is global to every ballot in a precinct, it allows the same ballot to be cast as many times as desired.

We demonstrate how this vulnerability can be very easily exploited by any voter during the actual election if she is allowed to operate the machine without being observed by a poll-worker. See Figure 9 for an example of an AV-OS ballot with the two Post-it notes affixed to its side. The attacker in this case is allowed to use the machine while unattended and he can pull out and re-insert the shown ballot so that the same vote is cast multiple times.

## 5 Recommendations for Safe Use

Given that the AV-OS terminal is merely a “bubble sheet” counting device and not a DRE system, there is no fear that the actual votes will be lost (since they are preserved in the voter generated paper trail). Nevertheless, the fact that the votes are not lost does not necessarily imply that they will be counted correctly. The attacks presented herein suggest that the AV-OS system has serious security defects in its design that demand strict observance of safe use guidelines. Based on our findings we propose the following:

- It is important to seal in a tamper evident fashion not only the memory card slot but also the serial port and the phone jacks of the terminal. Instead of sealing these sockets it is possible to disconnect them internally from the motherboard so that they are disabled as shown in Figure 10, although this approach has the disadvantage that its implementation cannot be verified without opening the system box.

Protecting the device with tamper-evident seals to secure it against opening of the system box is equally important. Opening the system box of the device not only makes the memory card exposed but also enables one to circumvent sealed serial ports by directly connecting a properly configured cable to the motherboard.

A complete approach involves protecting the entire AV-OS device within a tamper-evident enclosure at all times other than the actual deployment in an election. (This approach is being taken in Connecticut where the system carrying case is secured by a tamper-resistant numbered seal, with the seal number checked at all transit points.)

- Chain of custody should be strictly observed, from the point of initialization of the terminal to the time it returns to long-term storage after an election. The procedures for transporting and handling the equipment must be defined in advance (such procedures are in place in Connecticut).
- The memory card should never be allowed to be outside the AV-OS terminal (in fact this is the approach taken in the State of Connecticut). Given that no cryptographic integrity check is employed by the AV-OS memory card management, the moment the card is removed from a terminal it can be considered to be compromised. Unfortunately even if the memory card is sealed in the terminal, as our attack demonstrated in §4.1, the system does not guarantee an uncompromised election unless the remaining ports and case of the terminal are sealed in a tamper-evident fashion as well. While not directly addressed in this report, it is also necessary to safeguard the firmware chip in the AV-OS system. The chip contains the AccuBasic interpreter, and it is designed to be replaceable due to version changes (such as the change from 1.94.\* to 1.96.\*). It is imperative to ensure that the right chip is installed prior to the AV-OS machines being deployed for election.
- With proper precautions, the re-voting attack (cf. §4.2) should not be possible. The ballot should be filled out in private in an area separate from the ballot box, but inserted in public, while preserving the secrecy of the vote (several states, including Connecticut already use this approach). The AV-OS ballot

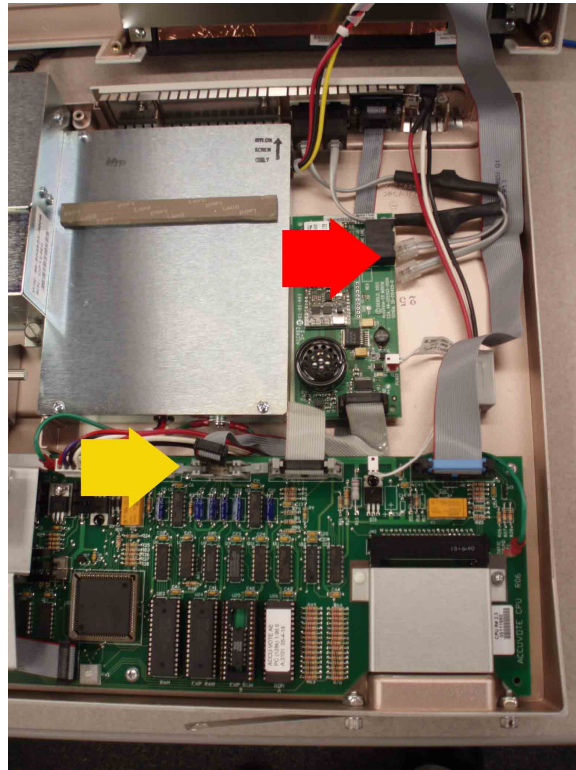


Figure 10: Disabling the serial line and modem of a GEMS terminal. The upper arrow (red in color layout) shows the disconnected telephone cables from the internal modem, whereas the lower arrow (yellow in color layout) shows the disconnected serial cable from the motherboard.

box should be kept under close supervision by poll workers during elections. Poll-workers must not be allowed to take their eyes off the machine, and should be wary of attempts at distraction.

- Finally, a post-election random audits involving hand counting of the ballots are highly recommended. (Such random audits will be conducted in the State of Connecticut.)

**Acknowledgments.** The authors thank the Office of the Connecticut Secretary of the State for supporting the production of this report.



## References

- [1] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin and Dan S. Wallach, Analysis of an Electronic Voting System, IEEE Symposium on Security and Privacy 2004, IEEE Computer Society Press, May 2004.
- [2] Ariel J. Feldman, J. Alex Halderman, and Edward W. Felten, Security Analysis of the Diebold AccuVote-TS Voting Machine, September 13, 2006  
<http://itpolicy.princeton.edu/voting>
- [3] Harri Hursti, Critical Security Issues with Diebold Optical Scan Design, Black Box Voting Project, July 4, 2005  
<http://www.blackboxvoting.org/BBVreport.pdf>
- [4] Harri Hursti, Diebold TSx Evaluation, Black Box Voting Project, May 11, 2006  
<http://www.blackboxvoting.org/BBVtsxstudy.pdf>
- [5] Susan Pyncheon, The Harri Hursti Hack and its Importance to our Nation, Florida Fair Elections Coalition, January 21, 2006. <http://www.votetrustusa.org>
- [6] Theodore T. Tool, MIT Guide to Lock Picking, 1991  
<http://people.csail.mit.edu/custo/MITLockGuide.pdf>
- [7] David Wagner, David Jefferson and Matt Bishop, Security Analysis of the Diebold AccuBasic Interpreter, Voting Systems Technology Assessment Advisory Board, University of California, Berkeley, February 14, 2006.



State of Connecticut  
Official Absentee Ballot  
Town of Westport

Be sure to read instructions on reverse side of this ballot.

OFFICE  
1 2  
Board of Finance  
Vote for Any Four

PARTY  
1A 2A  
R. Gavin S. Thomas C.  
1B 2B  
Steven L. Kevin A.  
1C 2C  
SAVE WESTPORT

State of Connecticut  
Official Absentee Ballot  
Town of Westport

Be sure to read instructions on reverse side of this ballot.

OFFICE  
1 2  
Board of Finance  
Vote for Any Four

PARTY  
1A 2A  
R. Gavin S. Thomas C.  
1B 2B  
Steven L. Kevin A.  
1C 2C  
SAVE WESTPORT

State of Connecticut  
Official Absentee Ballot  
Town of Westport

Be sure to read instructions on reverse side of this ballot.

OFFICE  
1 2 3  
Board of Finance  
Vote for Any Four

PARTY  
1A 2A 3A  
R. Gavin S. Thomas C. Ralph  
1B 2B 3B  
Steven L. Kevin A.  
1C 2C 3C  
SAVE WESTPORT

State of Connecticut  
Official Absentee Ballot  
Town of Westport

Be sure to read instructions on reverse side of this ballot.

OFFICE  
1 2  
Board of Finance  
Vote for Any Four

PARTY  
1A 2A  
R. Gavin S. Thomas C.  
1B 2B  
Steven L. Kevin A.  
1C 2C  
SAVE WESTPORT

State of Connecticut  
Official Absentee Ballot  
Town of Westport

Be sure to read instructions on reverse side of this ballot.

OFFICE  
1 2  
Board of Finance  
Vote for Any Four

PARTY  
1A 2A  
R. Gavin S. Thomas C.  
1B 2B  
Steven L. Kevin A.  
1C 2C  
SAVE WESTPORT

State of Connecticut  
Official Absentee Ballot  
Town of Westport

Be sure to read instructions on reverse side of this ballot.

OFFICE  
1 2 3  
Board of Finance  
Vote for Any Four

PARTY  
1A 2A 3A  
R. Gavin S. Thomas C. Ralph  
1B 2B 3B  
Steven L. Kevin A.  
1C 2C 3C  
SAVE WESTPORT

State of Connecticut  
Official Absentee Ballot  
Town of Westport, Connecticut  
Municipal Election

Be sure to read instructions on reverse side of this ballot.

OFFICE  
1 2 3 4 5 6 7 8 9  
Board of Finance  
Vote for Any Four  
Board of Education  
Vote for Any Three  
Board of Assessment  
Appeals  
Planning  
Commission

PARTY  
1A 2A 3A 4A 5A 6A 7A 8A 9A  
R. Gavin S. Thomas C. Ralph Charles W.K. Edward M. Lewis D. Garson F. Sean M.  
1B 2B 3B 4B 5B 6B 7B 8B 9B  
Steven L. Kevin A. Mark H. Mary R. Eleanor S.  
1C 2C 3C 4C 5C 6C 7C 8C 9C  
SAVE WESTPORT

Figure 11: The collection of the ballots used for the test election on the compromised AV-OS. Candidate “Thomas C.” receives 7 votes and candidate “Kevin A.” receives 5 votes.

```

PRECINCT 1
VERSION: 14      COPY: 0
COUNT: 1      SIZE: 128
ACCU-VOTE RELEASE: 1.96.6
REPORT:      USMA 1.2
PRECINCT CHECK: 7038
COUNTER CHECK: 290

TIME: 06:48:50  10/19/06

*****
** PRECINCT:    10 **
**              1
*****
BALLOTS CAST
20
*****
BOARD OF FINANCE
RACE # 30

BLANKS 7
R. GAVIN S. [REDACTED] 4
THOMAS C [REDACTED] 5
RALPH [REDACTED] 6
CHARLES [REDACTED] 4
STEVEN L [REDACTED] 7
KEVIN A [REDACTED] 7
# WRITE-INS 0
*****
BOARD OF EDUCATION
RACE # 40

BLANKS 5
EDWARD M [REDACTED] 3
LEWIS D [REDACTED] 5
MARK H [REDACTED] 0
MARY R [REDACTED] 7
STEPHEN M [REDACTED] 4
ROBERT HALE [REDACTED] 4
ROBERT M [REDACTED] 2
# WRITE-INS 0
*****

```

Figure 12: The outcome of the election as reported by the compromised AV-OS terminal. The votes of candidates “Thomas C.” and “Kevin A.” has been swapped. Candidate “Mathias H.” appears to have received no votes.