FRANK ASKIN, Esq.
PENNY M. VENETIS, Esq.
RUTGERS CONSTITUTIONAL LITIGATION CLINIC
123 Washington Street
Newark, New Jersey 07102
(973) 353-5687
Attorneys for Plaintiffs

Assemblyman Reed Gusciora, Stephanie Harris, Coalition for Peace Action, and New Jersey Peace Action, Plaintiffs, v. James E. McGreevey, Governor of the State of New Jersey (in his official capacity) and Peter C. Harvey, Attorney General of the State of New Jersey (in his official capacity), Defendants.	SUPERIOR COURT LAW DIVISION MERCER COUNTY Docket No. CIVIL ACTION

ANDREW W. APPEL, being of full age, hereby certifies:

1. I am a Professor of Computer Science at Princeton University, and a resident of Princeton Township, Mercer County, New Jersey. I received a bachelor's degree in physics, *summa cum laude,* from Princeton University in 1981, and a Ph.D. in computer science from Carnegie Mellon University in 1985. I have been on the faculty of

Princeton University since 1986. I attach my C.V. to this report as exhibit A.

2. My areas of expertise within the field of Computer Science include computer security, software engineering and design, programming languages, computer architecture, operating systems, and other areas. My primary research over the past decade is in software security: on what basis can we decide whether to trust the correct and safe operation of computers and computer programs.

3. I have studied the technological issues connected to the use of voting machines, and also the social and political context in which these machines are used. In the fall semester of 2004, I am currently teaching an undergraduate course at Princeton University on these topics. My research and teaching includes the study of a wide variety of voting technologies, including paper ballots, optical-scan ballots, punch-card ballots, direct-recording electronic machines, and other technologies including internet voting protocols. With each technology I study questions such as, “what protocols and safeguards are used with this technology, how effective are the safeguards, and what was the historical context that led to the introduction of these safeguards?”

4. In this report I will discuss direct-recording electronic (DRE) voting machines without a voter-verifiable paper trail, such as the machines that are to be used in many New Jersey counties in 2004. I will address primarily the question of, “can we be sure that the votes as the voter cast them are accurately counted using DRE machines?” However, before I discuss DRE machines I will discuss, for purposes of comparison, hand-counted paper ballots.

5. One of the most interesting problems in the design of protocols for elections in a democracy is that there are inherent conflicts of interest: partisan elected officials supervise the elections, or appoint those who supervise the elections, that elect those very officials. That we can still trust the fairness of the vote-counting is remarkable; but this trust comes in part from specific safeguards that are put into the process. One of the most important safeguards is transparency: both the public and the parties are invited to observe key parts of the process.

6. As I will explain, when DRE voting machines are used, that transparency is lost when a key step of the process—recording the voter’s vote into an electronic “ballot

box”—is done by software, out of sight of any observer (even of the voter himself).

7. How can we be sure that paper ballots as the voter cast them are accurately counted? History provides many examples of corrupt practices associated with paper ballots, such as the stuffing of ballot boxes, the replacement of ballots before counting, or the altering of ballots during the counting. Even the layman can observe several common practices that are now used in connection with paper ballots that have obvious purposes to prevent and deter fraud, including the following measures:

   a. Blank ballots are provided by the polling officials, not by the voters, and the supply of blank ballots is rigorously controlled. This helps to prevent extra ballots from being injected into the process.

   b. Members of the public and representatives of the parties and of the candidates are permitted to inspect the ballot box just before the polls open, to verify that it is empty. (I will henceforth use the term “challengers” for “representatives of the parties and of the candidates”.)

   c. The ballot box stays in full view of both pollworkers and challengers, so that they can see with

their own eyes that each voter inserts only one ballot and that no other tampering takes place.

d. At the close of the polls (in some jurisdictions) the ballots are counted on-site. Each ballot is inspected by a pollworker and by challengers, and the tallies are made by a pollworker observed by challengers.

e. The tallies for each district are posted locally. Challengers and the news media may take notes of the tallies, so that they may make their own independent totals of the tallies from all the districts.

f. The ballots are impounded and sealed after the election so that a recount may take place if the results are in dispute.

8. There are many other such safeguards, which I will not enumerate here. The point is that even if a candidate or a voter does not trust any individual official (or does not trust the entire political party that appoints the poll workers), the candidate's challenger can see for himself or herself whether or not there is tampering with the election.

9. A DRE machine consists of hardware and software. The hardware includes input devices (buttons or a touch

screen) that the voter uses to indicate a vote; output devices (lights or a video display) that the machine uses to confirm the vote to the voter; a memory, for recording the votes; and other output devices, for reporting the vote totals after the polls close. These latter output devices may be printers, removable memory cartridges, network connections, or other devices.

10. The DRE ballot is laid out so that, to the layman, there is an intuitive connection between the candidate's name (shown on a printed ballot sheet or a video display) and the input device (button or spot on a touch-screen) that one touches to vote for that candidate. However, in the hardware of the machine there is no particular natural connection. It is entirely at the discretion of the software to read the inputs from the voter, to indicate feedback to the voter, and to add to a particular total of the memory.

11. I will use the terminology "button" to indicate a mechanical switch (such as on the Sequoia Advantage machine) or a spot on a touch screen; and I will use "indicator" to mean the mechanism (a light or a spot on a video display) whereby feedback is provided to the voter of his or her choice.

12. Because there is no inherent internal connection between the buttons, the indicators, and the totals kept in memory, faulty software could very easily add a number to the wrong total when a button is pressed, or make some other error, thereby misrecording a vote.

13. Because the recording of the voter's intent is entirely at the discretion of the software, we (the public, the voters, and the candidates) must be able to trust the software. How can we know that the software will accurately associate the right button-pushes with the right total-column?

14. **Can fraudulent elections go undetected?** My chief concern in writing this report is to discuss whether it is possible for fraudulent election machinery to miscount the votes in election after election without ever being detected. In a system with paper ballots, there is physical evidence that may be examined. For example, in the disputed 2000 Presidential election in Florida, much evidence came to light, and the political and judicial processes could respond to the evidence. But a DRE machine leaves no independent evidence of the individual votes.

15. For this reason, as an expert in computer security, I strongly recommend that if DRE machines are to be used, they should be equipped with equipment to print a voter-

verified paper ballot. That is, after the voter makes his or her selection, the machine prints a record of the voter's choices. The voter can inspect this record, and then it is automatically deposited in a ballot box. There are two important principles: the voter's inspection of his or her own ballot is unmediated by the computer (the voter can see the printed paper directly), and the paper ballots can be recounted, either routinely or if there is suspicion of the machine.

16. If we are to use DRE machines without voter-verified paper ballots, as is currently the plan in New Jersey, then we must be absolutely sure that the software and hardware of the voting machine accurately count the votes. Assuring this is a difficult task, and I will discuss the procedures, difficulties, and limitations.

17. In the field of software engineering and computer security, there are three main avenues to the achievement of well-behaved software: (a) testing the software, (b) running the software in connection with hardware protection mechanisms, and (c) formal inspection and reasoning about the software. All three are useful and necessary in the context of computerized voting machines, but each has substantial limitations.

18. **Testing**. One could test the machine in public by casting a certain number of ballots for each candidate, printing the totals, and verifying that the totals are correct. This is an example of “black box testing,” that is, testing done without knowledge of the internal working of the machines being tested; it would be the kind of test performed by an election official who has not been given access to the source code (the human-readable program) running inside the machine. Black-box testing is often useful in catching some kinds of programming mistakes.

19. However, black-box “logic and accuracy” testing cannot reliably guard against fraudulent software: because the hardware of a voting machine is usually equipped with a real-time clock that is readable by the software, it is possible for the malicious author of fraudulent software to program it to behave properly on any day except the date of the election. Even in the absence of a clock, or even if the clock can be reset to simulate election day for the black-box test, it is possible to program the software to recognize other conditions that will only occur on election day. For example, the trigger for rigging the vote could be the number of ballots cast (a number that might not be reached in a black-box test), or a particular unusual combination of votes for obscure candidates on lower parts

of the ticket, or a write-in vote. It is extremely difficult to set up a black-box test that can mimic election-day conditions so perfectly that a clever (fraudulent) software design could not recognize the difference.

20. **Protection mechanisms.** Many computers are equipped with protection mechanisms that can prevent unruly software from performing certain operations. These protection mechanisms are very useful. For example, in a computer where two software programs are running, protection mechanisms can prevent flaws in program A from corrupting program B. As such, protection mechanisms are a useful part of a hardware/software system design. However, although they can guarantee safety, they cannot guarantee correctness: flaws in program A will be less likely to affect the operation of program B, but program A can still compute the wrong answer.

21. **Inspection and analysis of the software itself**. A computer program is an organized sequence of instructions that tell the computer hardware what operations to perform in what order. Programs are usually written by humans to be read by machines. Programs can also be read by humans. Because computer hardware is (generally) quite reliable and (generally) executes the instructions in a deterministic

way, in principle it should be possible for a person skilled in computer programming to predict what the programs will do when executed on the computer. Thus, for example, if the program in a voting machine does not accurately total the votes, one might think it should be possible to see that by reading and understanding the program. I will explain why this is difficult even for experts.

22. **Complexity of software**. Some computer programs are concise, clear, well-organized, and understandable. Unfortunately, in my experience teaching programming, I find that writing such programs does not come naturally to many people—not just beginners but practicing professionals—and many computer programs turn out to be huge, complex, ill-organized, and extremely difficult to fully understand. This problem is exacerbated by several factors. Computer software, once written, is often modified time after time, year after year. Even talented programmers can face deadline pressure and conflicting demands that cause them to make compromises in the clarity of the programs they deliver. Computer programs are constantly being adapted to new circumstances and modified to fix bugs. Very often the programmers modifying ("maintaining") the program are not the ones who wrote it

originally. The very process of having many people modify the program over many years can cause the program to become ill-organized and incoherent.

23. Computer programs are often very long. For example, the software in the Diebold AccuVote-TS machine is approximately 50,000 lines of text (3,000 pages). To analyze such a program takes much longer than simply reading it carefully. Every possible interaction between one line of the program and another must be considered, almost as if it were a mathematical puzzle.

24. Here and in some other places in my report, I use the example of the Diebold AccuVote-TS machine. I understand that this machine is not in use anywhere in New Jersey. However, it is the only DRE machine for which it is possible for the public to inspect the software program used in the machine, and it is similar in many respects to other DRE machines in use in the State. Here, I cite the size of the Diebold source code just to indicate the approximate size of a representative voting-machine program. Below I will specifically address machines used in New Jersey.

25. **Imperfection of testing and inspection**. Reputable computer software companies such as Microsoft often have great difficulty producing 100% accurate software.

Microsoft (and similar companies) have every incentive to produce correct programs, because flaws in their programs cause embarrassment to the company (and may drive customers away) and expense (in distributing “patches” and updates to fix problems). Such companies employ many means to try to produce bug-free programs, including software engineering standards (i.e., guidelines for programmers to follow), extensive testing, code reviews (i.e., reading the program carefully line-by-line and discussing it), and so on. Even so, many flaws slip through all of these processes and survive in the programs sold to customers; the evidence is in all of the patches and updates that the companies must issue to correct problems.

26. The experience of commercial software is instructive in another way about the limitations of testing and inspection. Program bugs slip through the very extensive testing and inspection that companies do. Once the software reaches users, some (malicious) users are motivated to do a much more detailed and intricate analysis. These “hackers” find bugs that they can exploit to produce viruses. For example, many computer viruses are able to spread because hackers find and exploit bugs that Microsoft’s own professionals were unable to spot and correct. This is not entirely Microsoft’s fault: it

illustrates the difficulty in making a complex artifact such as a computer program 100% reliable. It also illustrates that, based on our experience with many kinds of commercial software, it is unwise to expect any particular small set of testers and inspectors to catch every flaw.

27. **Deliberately deceptive programs.** Most importantly, the programming bugs that slip through the processes I describe in the previous paragraphs are generally honest mistakes by well-meaning programmers. It is much more difficult to catch deliberate "Trojan horses" put into programs by dishonest programmers, because the programmers can use many techniques to disguise their tracks. Let me give an example. Suppose I wish to make a voting-machine software throw elections to the Democratic candidate. I could write a line of the program that says, *"if it is a presidential race, and the candidate's party is 'Republican', and the day of the week is 'Tuesday', and the month is 'November', and if the time is between noon and 5:00 p.m., then add 1 to the other candidate's total instead of this one."* Of course, if I write the code just like that, then anyone reading that part of the program will see it and be naturally suspicious. So instead I would scatter the parts of this "hack" in different parts of the program. In the

random-number generator code (which tends to be obscure anyway, so that the reader won't notice that something is out of place), I'll put something to recognize where the Republican ballot line is: "if the third character of such-and-such a word is 'p', that set a certain variable to an odd number…" The point is that, if it is difficult to read and understand a program (and detect flaws) when the programmers are not deliberately trying to fool you, it's all the more difficult to understand programs that are deliberately misleading.

28. **Exploitation of nonmalicious errors.** A serious problem today in the field of computer security is that programming mistakes made by well-intentioned (nonmalicious) programmers can be exploited by malicious attackers. Not every program bug can be exploited this way, but it is often the case that the attacker can make use of the bug to gain total control of the program and modify it (even while it is running!) to behave in any way the attacker wants. Some of the bugs that have been identified in the Diebold software, for example, have this flavor: these bugs were (probably) not inserted by malicious or dishonest programmers at Diebold, Inc., but they could permit malicious outsiders to corrupt an election.

29. **What software is loaded?** Even if we could, hypothetically, write such clear and well-organized software that an inspector could say with certainty that it accurately counts votes, there is still a very serious difficulty: How can we know that the same software we inspect is the program that is loaded into the machine? A naïve method would be to have the computer print out the program onto an output device, but when we ask "the computer" to do this, we are really asking the software running on the computer to perform this action. If the software is fraudulent, it can be programmed to print out what the inspector expects to see. What criminal investigators do when they want to inspect the software on a personal computer is to remove the hard drive (where the programs reside) and use another (trusted) program to inspect the contents; they do not rely on the untrusted software to explain itself. An analogous inspection process would be necessary for each voting machine: whatever medium contains the software would have to be removed and inspected by an appropriate piece of equipment.

30. **Self-testing.** Similarly, there are limitations to what can be achieved by software that tests itself. Many voting machines have "logic and accuracy tests" (LAT) that are performed when the machines are turned on. In some

cases, these tests are programmed in software; in other cases, they require interaction with and mediation by the software. They can be useful in detecting malfunctioning hardware, low batteries, or (in some cases) program bugs. However, they cannot be relied upon to detect intentionally fraudulent software, for two reasons: First, they are part of the very software that might be fraudulent; and second, as I explained earlier, the fraud might be programmed to take place only at certain times of day.

31. **The malleability of software.** If a program is stored on a medium that is writable, such as an ordinary hard disk, or a RAM memory cartridge, then it can modify itself. This means that a fraudulent program can be programmed to throw an election, and then at 7:55 p.m. on election day, overwrite itself with a copy of the certified, nonfraudulent program. This property of software—the inherent erasability of the medium—is unlike mechanical machines or paper.

32. **Voting machines used in New Jersey.** I understand that the DRE voting machines used in New Jersey include the Sequoia AVC Advantage (a “full-face” machine), the Sequoia AVC Edge (a “video touch-screen” machine), and machines made by ES&S and Shoup. I will give specific opinions about some of these machines.

33. A “full-face” direct-recording electronic voting machine does not have a video display. Instead, the names of the candidates are presented to the voter on one large poster-sized piece of paper. Behind the paper display are several buttons (also called “switches”) and lights (also called “indicators”). To vote for a candidate, the voter pushes the button next to the candidate’s name, and the software is supposed to light up the corresponding indicator light to confirm the choice.

34. A “video touch-screen” machine displays the names of the candidates to the voter on a video display similar to that used on a notebook-sized computer. Because the screen is not (usually) large enough to display all candidates in all races at once, (usually) only one race at a time is presented to the voter. The voter indicates his or her choice by (depending on the machine) touching the screen where the candidate’s name is displayed, or pushing a button to the side of the screen nearest the candidate’s name.

35. I will now discuss the two Sequoia machines used in New Jersey. I have personally used one of these machines, and as I will explain, I have been shown the interior of the cabinet of one of these machines, whereby I was able to perform a superficial observation. In addition, I have

read two documents labeled as originating from Sequoia Voting Systems, Inc., of Oakland, CA. One is entitled "AVC ADVANTAGE SECURITY OVERVIEW" (copyright 1997-2004) and the other is entitled "AVC Edge Security Overview, Release 4.2" (copyright 1998-2003). These documents are attached as exhibits B and C, respectively. Both documents make statements of fact that are generally plausible and consistent with other knowledge I have of the machines, with one important exception.

36. The statement that I find clearly unsupported by evidence in Sequoia's AVC Advantage Security Overview (at 5) is, "In ten years, with over 12,000 AVCs in use, in countless elections and with countless numbers of votes cast, *not a single vote has been lost to equipment malfunction*." [italics theirs] An almost identical statement also occurs in the AVC Edge Security Overview. Let me explain why it is unlikely that there could ever be reliable evidence that this statement is true.

37. Whether the type of malfunction is a hardware error (such as an intermittently faulty switch), a software error (a program "bug"), or a deliberately fraudulent program, there is no independent record of the voter's choice; the only record is the one that the program itself records.

38. By contrast, with the (now infamous) punch-card technology, we know that in many cases “dimpled chads” occurred, and failed to be counted by the punch-card reading machines. I am not in any way advocating the use of punch-card voting. However, this illustrates an example of a case where the failure of one component of a voting system leaves physical, quantifiable evidence. In the case of DRE machines, there is often no trail of physical evidence of failure. Because of this special property of DRE machines, which are unlike any other voting technology in this respect, statements that “the machines have never lost a vote” must usually be treated as unsupported by evidence.

39. It is important to note that no independent tabulation, or “recount,” can be performed on the individual voters’ votes. None of the DRE machines now used in New Jersey print a contemporaneous voter-verified paper record of each vote. The machine does have the ability to print, at the close of the polls, record of every ballot cast (shuffled to preserve voter privacy). However, this record could be falsified by fraudulent or erroneous software, so it is not a truly independent check. It is not a “voter-verified” record, since it is not printed for the voter to see as he or she votes.

40. With the exception of this statement that “not a single vote has been lost,” in the two Sequoia “security overview” documents many of the statements of fact, if not all the statements of interpretation, are plausible. I cannot verify their accuracy. But in the remainder of this report I will provisionally rely on parts of these reports.

41. **Sequoia AVC machines.** Two different Sequoia machines are used in New Jersey, and I will compare and contrast their security features. The AVC Advantage is a full-face machine (ballot displayed on poster-sized paper), and the AVC Edge is a video display machine. They use different computers internally, and they use different software programmed in different programming languages. They have some security features in common, and some that are present only on the AVC Advantage, and lacking on the AVC Edge.

42. I have some familiarity and experience with these machines. On April 20, 2004 I used an AVC Advantage machine to vote in a school board election. On September 27, 2004 I visited the Mercer County Superintendent of Elections in Trenton, NJ and interviewed several employees about voter registration procedures and about Mercer County’s procedures for use of the Sequoia AVC Advantage machines, and I was able to observe an AVC Advantage machine.

43. The AVC Advantage has a front panel, for use by the voter; a side panel, for use by the election board worker, and a rear door, which is to be opened only before the polls open and after the polls close. I was able to inspect the inside of the cabinet of the machine, through the rear panel door, which was unlocked and opened for me. Inside the cabinet, visible only when the rear panel door is opened, is a metal computer enclosure, slots in which two memory cartridges are inserted (these were present and inserted when I viewed the machine), a printer, an emergency ballot box, and other components. I have also inspected a sample printout made from the machine's printer. The printer is not used while the polls are open; it is used after the polls close, to print vote totals for each candidate.

44. **Communication via cartridges.** Both Sequoia AVC machines communicate ballot data and results via the insertion of memory cartridges. The machine can read data from the cartridge (such as the list of candidates on the ballot, the "ballot definition"), and they can write data to the cartridge (such as the vote totals, the "results"). This mode of communication is a better design choice than communication through a network port. Other machines communicate via network connections to the telephone

network or the internet, which is now widely considered among experts to be a bad design choice, since it's harder to control access to the machine when it is connected to a network.

45. **Inserting fraudulent software**. One of the most important questions to ask about a voting machine is, "how easily could an unscrupulous person install new software in the machine?" This is of crucial importance, since the software is what decides how to interpret voters' button-presses. The Sequoia AVC Edge machine has substantially less protection against the insertion of fraudulent software than does the Sequoia AVC Advantage machine.

46. Designing a new (fraudulent) software program to be installed in a machine such as the AVC Advantage takes skill in computer programming, but not more skill than countless hackers around the world routinely demonstrate every year when they produce computer viruses. Once the program is designed, it would need to be installed in the voting machines. This installation does not require much computer-science sophistication, but the method would depend on the model of machine.

47. **Installing new voting software by plugging in a cartridge.** One of the most significant differences between the AVC Edge and the AVC Advantage is that the AVC Edge can

read a new software program from its removable cartridge, and the AVC Advantage cannot. That is, the very program that decides how to interpret voters' button-presses can be replaced by simply inserting a cartridge and typing a password. Professor David Dill, a computer scientist and voting-machine expert on the faculty of Stanford University, told me (October 12, 2004) that in 2003 he personally observed an election official in Santa Clara county, California demonstrate this procedure on the AVC Edge.

48. The two "security overview" documents from Sequoia are also instructive. The "AVC Advantage Security Overview" (at 12-13) devotes a full page to an explanation of the security measures that ensure that no program can be loaded into the machine through the cartridge, and that the machine can execute instructions only from its read-only memory (ROM). It is my opinion that the security measure, if it is as Sequoia describes it, should be effective for this purpose.

49. In contrast, the "AVC Edge Security Overview" is silent on this point. There appears to be no security measure that ensures that the computer in the machine can execute only from its read-only memory; of course, such a measure would be inconsistent with what Professor Dill

observed, that the machine *can* accept new programs from the removable memory cartridge.

50. Therefore I conclude that anyone with physical access to the AVC Edge machine for 5 minutes, and who knows the password, can install a new program into it. Password control has been observed to be weak in general in connection with voting machines. I believe that there are many ways that passwords could come to be known to many people who occasionally or frequently come in contact with the machine.

51. **Installing new voting software by tampering with the internals of the machine.** Although the AVC Advantage and the AVC Edge differ in whether new software can be installed through the cartridge, on both machines it is possible to install new software by replacing a ROM chip from the internal circuit board. The security features in the machines intended to prevent this can be circumvented. I will speak primarily about the AVC Advantage, but I expect that what I say is also applicable to the AVC Edge.

52. Removing and replacing the ROM chip is a routine operation requiring only simple tools. According to the manufacturer, the (removable) ROM memory chip inside the AVC Advantage's cabinet is covered by a numbered, tamper-evident seal. This seal is supposed to enable any

replacement of the ROM to be detected. This measure would work only if (a) an inspector knew to look for the seal, and (b) the seal could not be faked. I am an expert on computer science and computer security, not on physical tamper-evident seals, of which I have only a well-informed layman's knowledge. I imagine that since the technology of physical seals is literally thousands of years old, techniques have long since been developed to fake them. Therefore I will assume that it is possible that the ROM chip could be fraudulently replaced and that a counterfeit seal could be installed upon it.

53. It is likely that replacing a ROM chip and installing a counterfeit seal would require at least 10 minutes of unobserved access to the machine, and would require a bit more skill than simply inserting a memory cartridge and typing a password. Therefore I conclude that the AVC Advantage is not secure against replacement of its control software, although it is less insecure than the AVC Edge. If the control software is fraudulently replaced, then the machine could give votes to whichever candidate the program tells it to, regardless of the voter's choice.

54. **Detection of fraudulent software**. Fraudulent (or erroneous) election software can be installed in the machines either during their design and manufacture, or

after their manufacture. There are several means we have to detect such fraud (or error): automatic self-test, certification, and inspection.

55. **Self-test.** In general, self-test is not an effective defense against fraud. Sequoia's software performs security checks intended to ensure that the ballot definitions and the software itself are authentic. But there is an obvious fallacy in having the software in the machine certify its own authenticity; in effect, a fraudulent program would just say "yes, I'm authentic!" The only way to check the authenticity of the software in the AVC Advantage machine is to remove the ROM, and I will explain below why I doubt that this is being done by inspectors in New Jersey.

56. **Certification** is a process applied not to individual machines, but to an entire model or class of machines. This is done once, generally as a condition imposed by the purchaser of the machines. Its purpose is to verify that the design, and the implementation of the design, is sound, and that the machine will conduct fair elections. The computer program in each machine must be correct—especially on DRE machines with no voter-verifiable paper ballot, where we must put our entire trust in the computer program that runs on the machines.

57. The verification of computer program is very time intensive. Furthermore, the people who do verification of DRE software face a particularly difficult task because they must not only detect innocent errors, but they must be able to catch deliberately disguised misbehavior in the software. That is, someone who anticipates stealing an election has much to gain by deliberately writing software that will malfunction, and deliberately covering his tracks to mislead the inspectors. In this respect DRE software is similar to bank-machine software or gambling-machine software, where successful fraud can lead to substantial profit. Therefore it is advisable to certify DRE software using standards at least as strong as those used for bank machines and gambling machines, where computer security experts (that is, computer scientists with experience in understanding willfully fraudulently software) play an important role.

58. As even many laymen know, computer software companies are constantly modifying (and in some cases improving) the software they sell, to add new features or to fix bugs. For example, Microsoft publishes a new version of its Windows operating system about once a month. Makers of electronic voting machines do, from time to time (and in some cases very frequently) modify the software that they

use in the machines, and in many cases they upgrade (modify) the software in machines that they have previously sold. It is for this reason that the AVC Edge permits the installation of new software just by inserting a cartridge: it is to facilitate the process of upgrading machines already in the field.

59. Even when machines are not upgraded in the field, the manufacturer may make improvements or changes to the program in the machines they are selling. For example the Sequoia AVC Advantage was originally sold in 1987, and I understand it may have gone through some sort of certification procedure at that time. Sequoia's AVC Advantage Security Overview states, "The only significant change to the electronics in that time [since 1987] has been to increase the amount of memory in the system, both for more complex ballots, and to add new voting features." *This means that both the hardware and the software are not the same as what was certified in 1987* (if indeed it was certified in 1987).

60. Even the smallest change to a computer program—even a change of just one letter—can radically alter its behavior. It is entirely possible that program bugs (which could miscount the vote) or fraudulent modifications to the program could be inserted into "upgrades". Therefore it is

absolutely necessary that, if the manufacturer makes changes to the software, the new version of the software is subjected to the same scrupulous certification process that I described above. (Note, however, I have no reason to assume that the actual certification process that New Jersey uses is thorough enough to be adequate).

61. However, if the manufacturer makes frequent changes to the software—to fix bugs or to add new features—it is very cumbersome, slow, and expensive to recertify each version. Therefore, there are incentives for manufacturers to bypass the certification process and install uncertified software. It appears that this has in fact happened, according to news reports. "An audit of Diebold Election Systems voting machines in California has revealed that the company installed uncertified software in all 17 counties that use its electronic voting equipment."[1] Similar problems have been reported in other states.

62. Thus it is very important to know whether each new version of each voting machine (and each new version of its software) used in New Jersey has been recertified, and what the certification protocols include.

63. **How effective are certification procedures?** The question I address here is, can New Jersey's procedures provide any

[1] Wired News, December 17, 2003, at 1; wired.com/news/evote/0,2645,61637,00.html

confidence that the machines are trustworthy? I understand that the New Jersey law requires a “patent expert” and two “mechanical experts” to examine the machines. While this may have been adequate for mechanical lever-action machines used in New Jersey during most of the twentieth century, it is completely inadequate for electronic voting machines. A DRE has a few mechanical components, such as pushbuttons and a printer, but the vast majority of the components are electronic circuits that form what we call the “hardware” of the computer, and the computer program that we call the “software”. The overwhelming majority of the complexity of a DRE machine is in the electronic circuits and a computer software. Therefore it is absolutely necessary to have electrical engineers and computer scientists participate in certification—below, I will explain why even this may not be sufficient.

64. I can also draw conclusions from the experience of several other states, including Georgia, Texas, California, Maryland, Ohio, Florida, and other states. In these states, several manufacturers’ machines passed federal and/or state certification processes, including machines from Sequoia, ES&S, Diebold, and others. Then, in 2003, a copy of the Diebold AccuVote-TS machine’s software was leaked to a place where truly independent experts could

examine it, and at that point numerous very serious flaws were found in the software. That is, the only machine subject (inadvertently) to independent public review was found to be seriously flawed. Many states' certification procedures erroneously certified a hopelessly flawed voting machine. This does not give me confidence in the certification procedures used by the states and counties that adopted this machine.

65. **Inspection.** After a state or county has purchased a voting machine based upon a certification of its software design, it should periodically inspect the individual voting machines, to make sure they operate correctly. A machine may be inspected many times over its working life. Unlike certification, inspection need not include an expensive and time-consuming review of the software in the machine. Instead, it suffices to compare the software word-by-word to the software that was certified when the machine was purchased, to make sure that it matches exactly. This comparison can be done in seconds using standard equipment. However, there are two ways to do it, and one of them is inadequate. **It is inadequate** to ask the software, "are you the right software?" As I have explained above, fraudulent software can (in effect) just say "yes." However, almost all methods for inspecting the

software *that do not require removing nonvolatile memory (e.g., ROM) from the machine* are performed under the control of the very software that is being inspected, and therefore are equivalent to asking the software to inspect itself.

66. A better way of inspecting the software is to remove the media containing it from the machine, and use separate equipment to inspect it. This is how criminal investigators inspect PCs that are seized under search warrants: they remove the hard disk (the media that contains the software) and install it into another computer for inspection.

67. Therefore, to adequately inspect the AVC Advantage it would be necessary to remove the ROM chip from the machine and insert it into another piece of equipment to read it. However, I doubt that this procedure is employed in New Jersey, for the following reason: to remove the ROM chip requires destroying the tamper-evident numbered seal on it, and the re-installation of another seal. While it is certainly possible to do this, I assume that it is not being done. However, I have not had the opportunity to review the New Jersey inspection protocols, so I cannot be sure.

68. To adequately inspect electronic voting machines other than the AVC Advantage, a similar procedure would be required. In some cases it is not a ROM chip, it is a disk or a flash memory, but the principle is the same.

69. In addition to the software inspection, it would of course be necessary to check the correct operation of many other components of the machine, which I will not discuss in detail, except for one important component.

70. **Inspecting the software is not enough**. An electronic voting machine contains a computer chip that interprets the program stored on the ROM (or other media). If the computer chip itself is fraudulent, then it may execute a program of its own choosing instead of the one stored in ROM. This form of fraud would be difficult but not at all impossible to perpetrate. I'll illustrate by taking the AVC Advantage as an example, but the same principle applies to many other machines. The computer (processor chip) used in the AVC Advantage is the Z80, made by Zilog corporation. Zilog sells many varieties of Z80. The basic variety, used in the AVC Advantage, has no internal ROM memory, so it must execute programs from a separate ROM chip. That's good, because it means an inspector could, in principle, inspect the software in that ROM. However, Zilog also makes Z80's that contain RAM and ROM on the same chip with

the processor. It might be possible to substitute this version of the Z80 for the one that Sequoia uses. In that case, the computer would ignore the program in the ROM chip and run whatever program is directly loaded into the processor. Such a program could deliberately miscount votes. This kind of fraud could be done in a way that is extremely difficult to detect. *Thus there are reasonable scenarios under which it is practically impossible to be sure what software is running in a voting machine.*

71. **Optical scanning of absentee ballots**. On September 27, 2004 I interviewed Karen Howard, an employee of the Mercer County Board of Elections. One of her duties is to operate the optical-scanning machine which counts absentee ballots in Mercer County. She told me that she can count about 1,000 ballots in 10 minutes using this machine.

72. Ms. Howard showed me the optical scanning machine. It comprises three main components. There is a standard personal computer running Microsoft Windows. Attached to it is a standard inkjet or laser printer capable of printing on letter-size paper. Finally there is an optical scanning machine with a hopper and an automatic feeder. The hopper appears to take letter-size paper, which I presume to be the format of an optical-scan absentee

ballot. All three components can comfortably fit on a tabletop the size of a standard desk.

73. I presume that the counting of the ballots is done by software that is installed on the personal computer and runs in the Microsoft Windows operating system. Almost certainly it must be the case that the printing of vote totals on the printer, and the display of vote totals on the screen of the computer, must be mediated by this software. Therefore, fraudulent software installed on the computer could misrepresent the results of the absentee-ballot count.

74. Ms. Howard told me that the public is invited to view a test of the absentee-ballot counting machinery conducted about a week before the November election. The machinery is tested by feeding a number of ballots through the machine, and comparing the results to a hand count of those ballots.

75. However, this kind of test can catch only unintentional errors, such as malfunctioning optical-scan sensors or bugs in the program. But a fraudulent program could be designed to misbehave only on election day, since it certainly has access to the date/time function available from the Windows operating system. It could be programmed to work perfectly

on any day except the first Tuesday in November between 8 p.m. and midnight, and cheat only during that time.

76. Because the PC with the optical-scanning software sits year-round in an office in the Board of Elections, it is entirely possible that at one time or another it is unattended. The software on a PC can generally be changed simply by inserting a disk into the CD-ROM drive. Therefore it would be possible for an unscrupulous person to install fraudulent vote-counting software.

77. Furthermore, if this machine is ever connected to the Internet (as PCs routinely are these days), an unscrupulous person could replace software on it remotely through the network connection *without ever being in the room*, because of insecurities in the Windows operating system. Thus, even if we perfectly trust the integrity of all the employees of the Board of Elections, fraud could be perpetrated and perhaps never detected.

78. Fortunately there is a rather simple solution. Unlike votes cast on DRE machines, which cannot be recounted, it is possible to recount optical-scan ballots by hand. It would not be necessary to recount every absentee ballot; it would suffice to recount a randomly selected sample of the municipalities in the county. The recount would not need to be done the night of the election; it could be done

within a few days after the election. If there had been systematic miscounting of absentee ballots by the software—even if that miscounting were programmed only to occur on election day itself—it would (very likely) be detected by a recount of this form.

79. Unfortunately, this kind of simple solution (which can be used for optical-scan ballots), does not apply to the DRE machines used at the polling places in New Jersey. There are no paper ballots that can be recounted by hand. Only if these machines were modified to produce voter-verified paper ballots would such recounts be possible.

I certify that the foregoing statements are true. I am aware that if any statements are willfully false, I will be subject to punishment.

Dated: October 14, 2004
Princeton, New Jersey

_____
Andrew W. Appel