My name is Matt Bishop. I am an associate professor in the Department of Computer Science at the University of California in Davis. I do research in computer security, especially in the area of vulnerabilities. I have been involved in numerous analyses of the security of systems. I was part of the group that RABA Technology's Innovative Solution Cell assembled to perform a "Red Team" exercise to discover vulnerabilities in the voting systems that were to be used in the State of Maryland. We found many.

In my view, the key factor in the question of whether to use DRE systems is: do DRE systems add to the set of existing vulnerabilities in the election process? The election process is vulnerable with or without DRE systems. Dishonest people can rig or steal elections that use non-electronic technology such as punch cards. But there are procedural mechanisms in place to protect against these thefts. For example, in California, the optical scanners are validated by hand counts of some portion of the votes. Observers can watch this process. When ballot boxes are moved to the clerk recorder's office, observers can ride along to be sure the boxes are not switched. The only part of the election that cannot be observed is the individual citizen voting in the booth. So this procedural mechanism, providing the opportunity for the public to watch each step of the election process, does well in keeping elections honest.

DRE systems do not offer this same opportunity. The problem is that the public cannot watch and evaluate each step of the development and implementation of the DRE systems. Worse, there is no proof that the DRE systems work correctly. There is evidence, but the evidence is far from convincing. Let me explain this in more detail.

Underlying every computer system, including DREs, is a set of assumptions. The assumptions may be as simple as trusting the users not to alter information, or as complex as trusting a set of programs to work correctly under all circumstances. In computer security, we analyze systems to find the assumptions they are making, and then ask, "What if this assumption is wrong?"

As an example, the Diebold DRE system assumes that the order in which the candidate names are loaded onto the ballot in the DRE is the same as the order of the candidate names on the server on which the votes are counted. But what happens if the order is not the same? Then George Washington, candidate #1 on the server, would get the votes intended for John Adams, candidate #1 on the DRE ballot. So, we ask if it is possible to switch the order of the candidates, or if the software can be tricked into doing this.

As another example, that DRE system assumes that only authorized maintainers of the system would enter administrative mode on the DRE. The vendor locked the connector for the keyboard in a compartment on the system. The assumptions are that only the authorized maintainers would be able to open the lock, and only the authorized maintainers would hook a keyboard up to the port to obtain administrative mode. But the lock took under 10 seconds to pick with an off-the-shelf lock picking kit, and a keyboard could be concealed in a long sleeved shirt. As a result, in our test, we were able to enter administrative mode in under 20 seconds, and could have switched vote totals between two candidates. This would not be detectable using any procedural mechanisms, as the aggregate total of votes would be unchanged.

In order to determine if a system is working correctly, we need to know the requirements, which articulate many of the relevant assumptions. We translate these requirements into specifications that the system must meet, and design the system. We then either prove mathematically or, more commonly, argue convincingly that the design meets the specification. Here's the first prob-

lem: what is “convincing” to you may not be “convincing” to me, and we both probably have different ideas than many of the public. So who do we have to convince? The second problem is the conditions under which the system is used. If those conditions do not match the ones in the specifications, your system may not do what you want. It would be like the old Groucho Marx line about a doctor inventing a cure for which there was no disease.

After all this, we need to build the system: program it and add the needed hardware. This step is not susceptible to mathematical proof because the task is simply too complex. So, we have to argue convincingly that the implementation matches the design. Here, we rely not only on the DRE software, but also on the underlying software that manages the computer: the operating system and its supporting software. Finally, the system is tested and fielded. But again, the testing is never complete; parts of the system simply are not exercised.

There are special techniques of design and development, called “high assurance”, that aim to provide the amount and quality of evidence for independent analysts to validate the system. I am not aware of the existence of any such evidence for DREs; indeed, they are built using standard software engineering techniques that do not offer this convincing evidence. I am aware of evidence gleaned through testing, but that evidence tests only the end product, and a small part of that, in limited ways. The development process must be far more rigorous than it appears to be, and evidence of high assurance must be public.

To summarize: the goal of a DRE is to record votes accurately, to protect voter privacy, and to provide a mechanism to enable the verification of both these facets of voting. The current state of the art does not provide these mechanisms. DRE systems are vulnerable to attack, failure, and inadvertent error. Because the only record of votes DRE systems have is in their memory, and that memory can be changed by attack, failure of the system, or inadvertent error, there must be some way to validate the results independent of the representation of the votes in the DRE systems. In California, that will require a trail that voters can use to check that their votes are accurately recorded, and that election officials can use to determine the results of the election.