Coda in the Key of F2654hD4

Hal Berghel, University of Nevada, Las Vegas

As the US presidential election draws near, many of us can expect to find ourselves face to face with an electronic voting machine. It's time to re-examine the integrity of these machines.

Remember that default DES key that was hard coded in the Diebold voting machines for many years? Despite revelations that shocked the voting public, computer scientists concerned with the systems security of Direct-Recording Electronic (DRE) voting machines were central to uncovering the now familiar Diebold debacle of 2005—when activist Bev Harris discovered and then posted the unprotected source code for the Diebold TS and TSx voting machines. She recovered the code from the Diebold website.[1] What unfolded was a fascinating chronicle of corporate irresponsibility, hubris, incompetence, political chicanery, and power politics—all wrapped up in a story befitting a good dime store novel. And the proverbial plot thickened when computer scientists got involved—at that point, things got downright ugly. As painful as it was for the computer scientists involved, the country is far better off for it.

STANDARDS VACUUMS

The professional computing community is very familiar with the role of standards. Well-known standards, such as ISO 17799 for IT security and the ISO 9000 series for management, establish guidelines and general principles that codify industry best practices. In some cases independent certification bodies are used to assure customers and the public of compliance. Although we all work with standards differently, we can agree on two things: first, standards provide a minimal assurance of integrity and quality, and second, wandering too far afield usually comes at a cost in terms of safety, reliability, performance, profitability, and credibility for the affected organization and its representatives.

There are standards for quality, safety, reliability, and so on, in industries related to food, drugs, military equipment, manufacturing, computer equipment, software, household appliances, floor coverings, and paint, just to name a few. However, one area that's historically been immune to reasonable standards is the manufacturing and use of voting equipment—that which determines our political future.

0018-9162/16/$33.00 © 2016 IEEE

EDITOR HAL BERGHEL
University of Nevada, Las Vegas; hlb@computer.org

The voting franchise has been operating in a standards vacuum for decades, and this vacuum extends well beyond voting equipment.[2] The Diebold story is just the tip of the iceberg.

We know that using the same encryption key for every transaction hasn't been an acceptable practice since the Caesar Cipher was popular in ancient Rome. Furthermore, since the mid-1970s, the DES algorithm was known to be vulnerable to brute-force attacks because of its short key length.[3,4] After DES was deprecated by the National Institute of Standards and Technology (NIST) and replaced by the Advanced Encryption Standard (AES), Diebold went on to hard code it into the source code—a willful circumvention of best practices for the sake of cost savings and expedience. That they then lied about it was an even greater betrayal of the public trust. All of this was possible because of the lack of both industry standards and accountability to the public. Further, Diebold carried on this way for an entire generation of voting machines.

Diebold's story is a shining example of the voting machine industry's heritage of stupidity and arrogance and the public's tolerance of proprietary electronics and software that have never been adequately tested by impartial, legitimate domain experts. Harris's disclosure broke the proprietary veil, and some daring computer scientists read the curious public in on some outrageous security breaches that we never would have known about.

Exposure of the Diebold AccuVote system's weakness is generally credited to Johns Hopkins University computer scientist Aviel Rubin and his colleagues, who in 2003 began analyzing the source code discovered by Harris.[1] It's useful to frame this story in terms of Rubin's analysis of the Diebold source code. Here's what he found:

1. The AccuVote system anonymized the voting order with a linear congruential generator (LCG) that didn't work properly and was inappropriate for this purpose, thereby undercutting the principle of the secret ballot.
2. In parts of the code that required cryptography, either the algorithms were incorrectly applied or not used at all. (Of course, as mentioned, DES was used despite having been deprecated by NIST.)
3. Diebold's approach to key management was juvenile. The same encryption key (see column title) was hardcoded into every voting machine. The vulnerability stemming from a lack of key management was first reported in 1997 by University of Iowa computer scientist Douglas W. Jones without effect (http://homepage.cs.uiowa.edu/~jones/voting/dieboldacm.html).
4. The association between a candidate's record in the ballot definition file and the appearance on the screen of the AccuVote DRE voting machine wasn't cryptographically protected, thus verifying that a voter's intent was accurately recorded was impossible. The fact that a vote appeared in the confirmation screen (front end) was no guarantee that that vote was recorded and tabulated (back end). Diebold's "redundant" storage technique only ensured duplicate copies of the voting transaction, not that any of the transactions were correct.
5. The ballot definition file contained sensitive information like the terminal's ID number, the dial-in numbers for online tally reports, IP addresses for networked computers, and user names and passwords—all in plaintext.
6. The smartcards used by voters to authenticate with the voting machines used no cryptography at all. Therefore, anyone with the ability to create smart cards offsite and get them inserted into an AccuVote DRE station (trivial—see below) could authenticate to the machine and have votes recorded.
7. Election officials' administrative cards all had a default PIN of 1111.

The Diebold debacle is fascinating chronicle of corporate irresponsibility, hubris, incompetence, political chicanery, and power politics.

Note that these are professional, technical comments—not parochial or political opinions. Most of us recognize the faults as rookie mistakes that wouldn't withstand scrutiny in a respectable college-level computer science programming class. Rubin claims in his book that this level of sloppiness was characteristic of the entire Accuvote TS code base. For making these deficiencies known, he was vilified by Diebold, sundry election officials, and an occasional politician. Such blowback against technical experts was repeated several times before the Diebold story had played out. We'll pass over the idiocy of making unprotected source code available through Diebold's website in silence. The point to remember is that Rubin's was the same low assessment of source code that any of us would give to our students. It was an accurate, fair, and legitimate criticism of sloppy work.

However, Rubin wasn't the Diebold code's only critic. A series of investigations by computer security specialist, Harri Hursti, proved to be even more embarrassing.

THE HURSTI HACKS

In the mid-2000s, Hursti conducted several evaluations of the Diebold AccuVote system on behalf of Black Box Voting, Harris's election activist organization. The analysis was fairly extensive, pointing to deficiencies in the Diebold boot loader, removable memory, and easy-to-circumvent hardware security. Hursti correctly assessed Diebold's three-layer architecture as insecurity-in-depth, which is one step below security through obscurity.[5] Without question, the most alarming insecurity had to do with Diebold's removable memory cards. In the literature, these are the most prominent of the so-called Hursti hacks. These hacks were so simple and dramatic they were featured in the 2006 HBO documentary, *Hacking Democracy* (http://hackingdemocracy.com).

Hursti showed that the insecurities of the AccuVote TS and AccuVote TSx OSs were so substantive that even elementary changes to the code and/or data fields on the removable memory cards could change the outcome of elections. Despite the fact that these hacks had been demonstrated several times in several different jurisdictions, the initial response from Diebold was to attack the messenger(s). They even demanded that HBO cancel the previously mentioned documentary, but without effect.

Remember that Rubin published his analysis of the source code and Hursti followed up with experimental demonstrations of some result-altering hacks. Diebold's defense and counterclaims began to permanently unravel when University of California and then Princeton University researchers confirmed Hursti's results.[6,7] An earlier independent review by the Science Applications International Corporation (SAIC)—commissioned by the State of Maryland in 2003—had also reported that the "AccuVote-TS voting system is not compliant with the State of Maryland Information Security Policy and Standards ... and is at a high risk of compromise."[8]

A follow-up review by Maryland-based RABA Technologies, LLC, echoed the SAIC report, and added that the back-end management system (GEMS) was also insecure. So by the time *Hacking Democracy* came out, Diebold's proverbial cat was separated from the bag by light years. Diebold's response to these revelations was typical of the power elite: "... voters in the state of Maryland can now rest assured that they will participate in highly secure and accurate elections." Then-governor Robert L. Ehrlich Jr. (R), opined that "Because of this [SAIC] report, Maryland voters will have one of the safest election environments in the nation" (both quotes appear in *Brave New Ballot*,[1] pp. 137–138). There's no way to know whether the better explanation of Diebold's and Ehrlich's spin is cognitive dissonance or outright deceit, but whatever the reason, the known code insecurities went unattended for many years.

Although the hacks themselves are of only marginal historical significance at this point, the complex interplay among Diebold, the election officials who either tried to cover up the insecurities or expose them, the politicians who sought political cover from the exposures, and the computer scientists who uncovered the problems remains critical for little has changed to correct the problems. The capacity of the manufacturers, vendors, and election officials to conceal, cover up, and deceive is as great today as it was 15 years ago. But more is at stake now because DRE voting machines are ubiquitous, and we've since developed a tolerance for chicanery in our elections. There's another player that I haven't mentioned: the Independent Testing Authorities (ITAs) that "validate" these voting systems.

ITAs AND VOTER "VERIFICATION"

Diebold, Sequoia, and Election Systems and Software (ES&S) came to dominate the digital voting equipment market by the early 2000s. After a few mergers and acquisitions cycles, Diebold and Sequoia became subsidiaries of Dominion Voting Systems. At this point, the competition has been narrowed to a very few players.

Once a voting system is developed, election officials might be deluded into a false sense of security by ITAs (now called Voting System Testing Laboratories) that certify the system's integrity. ITAs work in much the same way as credit ratings services (think Moody's, Standard and Poor's, and the Fitch Group), and they're bound by the same incentives. In all cases, the applicant pays for the service of certification—that is, the beneficiary of the certification provides the revenue stream to the certifier. Thus, if an ITA rejects certification of voting equipment (not likely), other ITAs are enlisted until one approves certification. This arrangement takes conflict of interest to a new plateau.

Further, ITAs/VSTLs only do extensional validation, which is to say they compare results by re-running election records with known outcomes or using canned datasets with well-defined data. That doesn't really contribute much confidence in the system if no one looks "under the hood." Chip design and circuit analysis aren't part of the validation because both are proprietary. No objective source code review is undertaken by skilled computer scientists, unless there's been an accidental leak like the one mentioned earlier. This incestuous relationship among ITAs, manufacturers, and technically ignorant election officials seeking to avoid public scrutiny of their activities is still with us today. This is why the Diebold story is still relevant. Although Diebold Election Systems and its amateurish source code is gone, the structural problems that gave rise to them in the first place are still with us.

If an ITA is to be effective, it must not only provide intentional validation that compares output from canned datasets, but it must also provide a functional analysis of the source code by impartial, skilled computer professionals. This is in effect what Rubin's, David Wagner's, and Ariel Feldman's teams did.[6,7] But at this writing, verification of voting systems' code might amount to nothing more than comparing hash signatures between file versions. Hash signatures are measures of binary identity—not the quality or integrity of code. (Primitive analysis restricted to I/O based on canned datasets is frequently referred to as blackbox analysis, which is the source of the name Harris chose for her elections activist and investigatory group Black Box Voting.)

Further, Diebold apparently didn't bother to run their source code through a commercial-quality source code analyzer. However, Wagner and his colleagues did.[6] In the research they prepared for the California Secretary of State, the Fortify (now HP) static code analyzer identified 16 security vulnerabilities in the AccuVote operating system ranging from array bounds violations and faulty input validation errors to buffer overruns, buffer underruns, and pointer errors. Although the specific details (location of code fragment, and so on) were suppressed from the public report, enough detail was included to convince any computing professional that the code base lacked integrity. Note that these 16 vulnerabilities would have been easily detected with the HP Fortify static source code analyzer had Diebold chosen to use it. I encourage readers to review these referenced reports and confirm for themselves that the Diebold source code used up to and including the 2006 national elections should be on display in the Smithsonian as a primitive artifact. Any thoughtful analysis confirms the computer scientists' claims: the code was amateurish, the security standards were embarrassingly weak, and the systems were fraught with vulnerabilities. According to *Democracy Hacked*, approximately 40 percent of all votes in the US were counted by systems that ran this code at the time of the analysis.

The Diebold source code used up to and including the 2006 national elections should be on display in the Smithsonian as a primitive artifact.

DON'T WORRY, BE HAPPY

Some have claimed that DRE voting machine security isn't an issue:[9]

> *The conjecture that ... we are unable to make such a simple system secure and accurate is contradicted by the facts of our everyday existence. We build secure and accurate computer systems that fly our airliners. We build secure and accurate computer systems that guide our submarines under the ice cap. We build secure and accurate computer systems that guide our astronauts to the moon and bring them safely back to earth. We submit to open heart surgery while a computer monitors our vital signs and controls an artificial heart and lung machine. The list of secure and accurate computer systems that monitor, control, and improve our lives is large and growing daily.*

The appropriate response to this argument is "that's true, but so what?" This is a patently silly position to take for a number of different reasons. For one, threat vectors must be understood in context. Who'd be incentivized to corrupt flight control systems? What might be gained if an airliner was off course? What relationship would the possible perpetrators likely have to the affected airlines? The same applies to moon landings, heart monitors, and the like. In each case, the likely threat would be terrorists or criminals, and external. The fear would be that someone outside the system wants to do harm to others and the incentive might be revenge, anger, hate, jealousy, greed, and so on—all motives that are visceral and personal.

Voting machines provide an entirely different context: the incentive is to subvert the democratic process toward partisan effect, and the likely perpetrator would be internal, or at least very closely linked to a specific political interest. Thus, the perp would likely be a partisan operative either employed by, or closely connected to, a candidate, party, PAC, or particular election officials. Murderers and terrorists tend to not work closely with domain knowledge experts on their weapons of choice. People that steal elections do.

Further, mission-critical systems rely on high-confidence software development paradigms. As shown, the Diebold code certainly wasn't high-confidence. If loss-of-life scenarios require high-confidence methodologies, loss-of-country scenarios make them similarly desirable. The Diebold DRE voting machines under discussion weren't trusted systems, rather, they're twisted systems in which minimal attention was paid to best practices in software development, software security, user privacy, software reliability, and so forth. The letter and spirit of industry standards in effect at the time the equipment was developed were violated, ensuring that expected results would be obtainable only under optimal circumstances in which all involved behaved properly and predictably and without serious corrupting external influences. Elections never offer such controlled environments.

In addition, as long as completely secret ballots are required, there's no way to fully and incontrovertibly audit the DRE voting machines' results. Even when paper-tape backup is used, the voting sequences are scrambled so the individual votes can't be recovered, and there's no voter verification of the vote, but only voter verification of the most recent behavior of the particular voting machine. Although some alternate voting systems have been proposed,[10] no commercial voting machines that I know allow each individual voter to verify that their vote was actually recorded by the digital vote management system and reported to state election officials correctly. For that level of assurance to result with current voting equipment, both the electronic and physical records have to be tallied and publicly reported independently. It's important to understand that the phrase "voter verified" normally refers to the voter's verification of the vote cast at the DRE voting terminal, not verification that it was recorded by the tally management system.

Murderers and terrorists tend to not work closely with domain knowledge experts on their weapons of choice. People that steal elections do.

WAG THE DOG

In most important respects, little was learned from the Diebold fiasco. To be sure, DRE voting machine manufacturers are more attentive to source code integrity, but the degree is a matter of conjecture because the code is still proprietary and not available for inspection—not even after the election concludes. Lack of trustworthy code remains a real and present danger to election integrity and our democracy. Vilification of the computer scientists wasn't due to their scientific results—no reasonable person challenged their facts—but because they cast doubt on the accuracy and honesty of the election results. In other words, they were castigated for pointing out the obvious: no insecure computing systems are trustworthy, DRE voting machines included!

In addition, the few electronic voting machine manufacturers that remain are more circumspect in how they represent their product to their customers—the jurisdictions and the public. Although Diebold's arrogance and hubris waned as the company headed toward collapse, there's still no mechanism through which the public can establish confidence in these manufacturers' products and practices—too much is hidden from view.

Thus, we can't know the degree to which obsolete, insecure code and bad practices are in active use. It's also unknown whether, or to what extent modern source code is built around a valid security model. In 2016, vendors and manufacturers can still manipulate election officials who lack technical knowledge and skills. Voting machine procurement and approval processes face partisan brinksmanship, and any election official demanding independent testing faces threat of litigation from manufacturers and perhaps even election officials. The overwhelming majority of jurisdictions still fail to perform complete audits of election results, and what's more, there's evidence to suggest that those who scrutinize the fairness of elections might be subject to government surveillance.[11] These aren't good signs.

At its inception, digital voting technology promised to promote universal suffrage—enabling underprivileged, disadvantaged, and immobile voters to come to the polls, as well as related benefits such as the minimization of pressure from partisans and some mitigation against the historical vote-suppression techniques (such as long wait times, confusing ballots, miscounts, undercounts, discarded ballots, corrupted results, and so on).[12] However, DRE voting machines haven't delivered on the promise to help secure the election franchise for all citizens. The Diebold scandal revealed that there's far too much slop in the digital voting process at too many different levels. For that understanding, we're indebted to the computer scientists mentioned in the studies I've referenced here. They're the true heroes of this story, and the country owes them a great debt.

All in all, the way electronic voting is administered in the US still falls far short of reasonable expectations in terms of the ability to verify election outcomes;[13] to technically validate the equipment's source code; to achieve a public understanding of the systems' vulnerabilities; and to completely disclose possible conflicts of interest between vendors and their agents, public officials, and those engaged in vetting the integrity and certification of voting systems. In these areas, we're no farther along than we were 20 years ago. Our best hope at addressing these problems rests with computing professionals. Indeed, our goal should be to demand that these experts be centrally involved in vetting all future voting systems. After all, our first line of defense is the community of computing professionals who are willing to take risks and speak out.

For those interested in this topic, I recommend the seminal book, *Broken Ballots: Will Your Vote Count?*[20]

REFERENCES

1. A. Rubin, *Brave New Ballot*, Morgan Road Books, 2006.
2. H. Berghel, “Digital Politics 2016,” *Computer*, vol. 49, no. 1, 2016, pp. 75–79.
3. L. Hoffmann, “Q&A: Finding New Directions in Cryptography” [interview with Whitman Diffie and Martin Hellman], *Comm. ACM*, vol. 59, no. 6, 2016, pp. 110–112.
4. J. Gilmore, “DES (Data Encryption Standard) Review at Stanford University,” Toad Hall, 20 Sept. 2005 (with subsequent updates through 2015); www.toad.com/des-stanford-meeting.html.
5. H. Hursti, “Diebold TSx Evaluation—Critical Security Issues with Diebold TSx (unredacted),” security alert, Black Box Voting, 11 May 2006; www.blackboxvoting.org/BBVreportIIunredacted.pdf and Supplement www.blackboxvoting.org/BBVreportII-supplement-unredacted.pdf.
6. D. Wagner et al., “Security Analysis of the Diebold AccuBasic Interpreter,” Voting Systems Technology Assessment Advisory Board report, Presidential Commission on Election Administration, 14 Feb. 2006; https://web.archive.org/web/20070611092341.
7. A. Feldman, J.A. Halderman, and E. Felten, “Security Analysis of the Diebold AccuVote-TS Voting Machine,” *Proc. USENIX/ACCURATE Electronic Voting Technology Workshop* (EVT 07), 2007; www.usenix.org/legacy/events/evt07/tech/full_papers/feldman/feldman_html/index.html.
8. “Risk Assessment Report Diebold AccuVote-TS Voting System and Processes,” report commissioned by the Maryland Dept. of Budget and Management, Science Applications Int’l Corporation (SAIC), 2 Sept. 2003 [redacted at the request of the State of Maryland]; www.ballot-integrity.org/docs/SAIC_Report.pdf.
9. B. Williams, “Presentation to the US Election Assistance Commission,” presentation, 5 May 2004; www.eac.gov/assets/1/AssetManager/testimony%20brit%20williams%20kennesaw%20university%20public%20meeting%20may%205%202004.pdf.
10. D. Chaum, P. Ryan, and S. Schneider, “A Practical, Voter-Verifiable Election Scheme,” *Proc. 10th European Symp. Research in Computer Security* (ESORICS 05), 2005, pp. 118–139; www.pretavoter.com/publications/esorics05.pdf.
11. G. Howland Jr., “www.bigbrother.com,” *Seattle Weekly*, 9 Oct. 2006; www.seattleweekly.com/2004-05-19/news/www-bigbrother-gov.
12. T. Campbell, *Deliver the Vote: A History of Election Fraud, an American Political Tradition—1742–2004*, Carroll & Graf, 2004.
13. B. Schneier, “The Problem with Electronic Voting Machines,” *Schneier on Security*, 10 Nov. 2004; www.schneier.com/blog/archives/2004/11/the_problem_wit.html.
14. D.W. Jones and B. Simon, *Broken Ballots: Will Your Vote Count?*, Center for the Study of Language and Information, 2012.

HAL BERGHEL is a professor of computer science at the University of Nevada, Las Vegas and is an IEEE and ACM Fellow. Contact him at hlb@computer.org.