Chapter 10
Black Box Voting

Ballot Tampering in the 21st Century

by Bev Harris
with
David Allen

Edited by
Lex Alexander

Cover Art by
Brad Guigar

10
Gently now...Carefully...Take the Lid Off and —
Eeeeew!

This chapter will be delving into unavoidably technical areas. This presents a challenge to the reader if, like me, you don't have a computer background. Even if you don't understand the specifics of the flaws uncovered, the gist of the problem is apparent. You will see our evolution from curiosity, to concern, to alarm as we unravel the voting system.

I certainly am not a programmer and, aside from looking at filenames, I wasn't much help in analyzing what was in the files. But by June 2003, Diebold voting files had begun to pop up in various places, and we learned that citizens all over the world are deeply interested in how their votes are counted.

Spontaneously, people began analyzing the voting system files, discussing them, and doing a little surreptitious comparison of their findings. *Hey, come over here, look at this ... We're trying to find out how our vote is counted!*

"This is dangerous," someone explained, to everyone's surprise. "Bad things could happen. Very bad things."

Can someone please explain to me how our "democracy" turned into something where ordinary citizens can get arrested just for looking at how their votes are counted? No, I'm not asking you to explain the "Digital Millennium Copyright Act" (DMCA),[1] which in Internet circles is almost as controversial as the Patriot Act. The DMCA was designed to clamp down on music swapping, but somehow it morphed into a tool that can eliminate free speech without due process and may punish copyright violations with jail time.

Some people say the DMCA might be used against any citizen who studies the software that counts his votes. What I want to know is this: How can we call ourselves a democracy if we are so afraid of the consequences that we don't dare to inspect our own vote-counting system? No, don't take this opportunity to

describe the DMCA law to me, or explain the history of how this came to be — what I'm looking for is an explanation of how *scaring people* who simply want to make sure their votes are counted properly can possibly be the right approach to a robust democracy.

Apparently, this peeking at how we count votes is dangerous and (possibly) forbidden — but no one seems to know for sure. Lawyers confess to uncertainty as to whether looking at vote-counting files found on an open Web site can be permitted.

For several months, I considered this issue. As of the writing of this book, I've not yet been able to get a straight answer out of anyone. Here is what *I* came to believe, after much thought: I think that examining our voting machine software is not only a legitimate activity, but it is also our civic duty. For queasier souls, I offer these statements in defense of this endeavor:

1) These files were publicly available.
2) Examining them is in the public interest.
3) Our objective is study and review, not copying and selling voting systems.
4) In a democracy, vote-counting should not be secret in the first place.

The Internet is alive with message boards, chat rooms and forums. People go to these Web sites to meet and converse with each other, using whatever name they choose so that they can feel free to express any opinion they like. One such forum is DemocraticUnderground.com (DU), a rapid-fire political discussion board with more than 30,000 participants. Because this kind of venue provides a feeling of safety and anonymity, citizens have been able to muster up the courage to examine our voting system.

I perused more than 5,000 comments about voting systems from DU, and I think you'll agree that the excerpts from the 75 posts that follow show a remarkable picture of democracy in action.

> "I haven't seen the Diebold machines or how they operate, but in my precinct, we have a numbered ballot we fill out that is scanned into a machine. In case of a questionable result, the numbered paper ballots can be used to verify results by

* In order to protect the innocent from the guilty, we have changed all the screen names.

> a hand count. The Diebold machines should have something similar."
>
> — "Cleaver"*

Three months later, "Cleaver" got a rude awakening. He learned that he has indeed been voting on Diebold machines and that a security breach was discovered right in his home county.

A lively discussion took place when programmers began looking at the source code itself:

> "What could this thing possibly be doing to need so much source code? I have built systems ten times more complex than any imaginable voting machine in 1/100th the source code space. Sometimes when programmers don't know what they are doing this is the result – lots of cut and pasted functions that are almost the same, tons of obsolete but not removed code ... Ugh."
>
> — "Romeo"

"Given that professional programming is complex by its nature and professional programmers are often messy tasteless people by 'normal' social standards, I'd be surprised if it didn't look like this. In fact, while the sample in question is small, it looks like at least half of the source is visual C++ generated from templates by click&drag, by virtue of its unpleasant-to-type words.

"Once the compiler gets hold of it, chops logicals and optimizes loops, you'll never know how crappy the source looked anyway. Believe it or not, there are actually contests (such as the infamous 'obfuscated C contest') to write the most convoluted and inscrutable programs possible."

— "mortal"

> "I don't think it's likely that you can prove anything with the source code. You won't find a function called "double_GOP_Votes" that does fake counting ... nevertheless, we could very well find backdoors, which aren't that uncommon, that would allow tampering."
>
> — "BettaWatchYerVote"

Some participants argued about the discussion process itself...

> "The thought struck me after reading the third or fourth message that this dialogue should not be on a public forum."
>
> — "ErgoWeAre"

> "Why not? This is the very underpinning of democracy we're discussing here. If there was ever a need-to-know issue for the general public, this is it."
>
> — "mortal"

Others suggested the most efficient ways to hunt for vote fraud:

"Have any empirical tests been done? Meaning, generate a large amount of output with the code, and analyze that output, looking for anything the least bit funny, then going back and then focusing on those funny results to look for foul play."

— "Ovaltina"

> "**Ok, so you've got your haystack and you're looking for the needle** ... Here's how I'd approach this problem:
> "...I'd begin by doing a bit of analysis on how the system is structured. Isolate the important data types (that voter info one is a good example) that someone might be interested in modifying...
>
> "After that, I'd go a few levels deeper with the functions that are doing the data modifications (look at the functions that are called by those functions.) I'd begin to chart out the "life of a vote" in the system...
>
> "...[I'd look for] code that does not appear to do what it's comments say it's supposed to do; code that is completely undocumented; any code that seems to be manipulating memory in "weird" or unnecessary ways. God help you because this is in C++."
>
> — "Bibbidi-Bobbidi-Boo"

One participant began to explore new legal issues...

"Discussion cannot be considered illegal under the DMCA.

"By making this third party code available freely, Diebold was violating the DMCA. If you would like, I could compile a quick list of third party companies and the files they are responsible for. Just those company names alone could provide you with multiple avenues of research...It's unfortunate that Diebold allowed Microsoft source code to be publically available on one of their FTP servers."

— "Clark Kent"

User manuals began to surface, answering many questions about how to operate the systems but sometimes raising new areas of inquiry.

> **"Look at this sentence**: *When you have finished entering the totals for a precinct, all Check values must be zero in order for you to proceed to the next precinct. If necessary, you can make up the difference by putting the number in the Check tally in the Times Blank field if the race is a Vote For One race. If not, you may have to perform some additional calculations to make the Check value equal zero.*"
>
> — "Jolio"

"I'm a technical writer, and even *I* can't figure out if that says what we think it says or not. Enter that one in the STC's "Worst Manual of the Year" contest. "

— "Crapper Dan!"

As time went on, a note of concern entered some comments:

> *"Why are they entering manual votes? If we have optical scanners reading absentee and touch screens reading polling votes (and the touch screens also read the challenge votes) — what is the purpose of manual entry?"*
>
> — "Jolio"

"My guess the optical scan machines may not be integrated into the same computer system as they are using to run the GEMS software. so (i am guessing) the data has to be entered manually. Even [if] the optical scan machines WERE on the same computer, it might be necessary to enter the

data manually if there is no standard protocol for transferring the data from the "optical scan" app to the GEMS software. Another possibility is write-in votes or provisional ballots."

— "K3Park"

"Write-ins, provisionals handled on the touch screen and most systems with touch screens are integrated with optical scan systems, but not all. That could be the reason for it, but if so...what security measures should it have, at a minimum? Because, manual entry might have a legitimate purpose for entering absentee votes, yet provide a back-door for tampering also."

— "Jolio"

"Unfortunately, a key piece is missing, manualentry.cpp — It's documented, but is not there."

— "Clark Kent"

"That's right... The code for GEMS Server is the key and it ain't here. Look, there's the code for the soon-to-be-hundreds-of-thousands of touch screen stations, and then there's the code for the servers."

— "Rummage"

"The system has a history of 'space' problems:

- Fixed problem with Accumulator not working with large elections (out of space).
- Fix problem with removing system.bin and AVTSError.txt files when removing old election files to make more room on the storage device.
- Add checking for minimum storage space free before allowing a ballot to be cast."

— "Lucille Goldman"

"*They have had one hell of a time with standard magnetic card readers. Programmer frustration comments are rampant in this series of modules.*"

— "*BlueMac*"

"It took 'em three years to log manual entries ... sheesh!
• Fix problem with wrong time being stored in the audit log.
• Add log entry for posting of manual results"
— "Lucille Goldman"

"I see the section on manual entry. Not a word in it on who is allowed to do it — presumably, must be someone with admin privileges, but I note this manual also has a section for remote access to the database (why does any election supervisor need to remote access their computer for voting program tasks?)

"And uh — wouldn't you say that a key event to log [in the audit] *after launching the election would be to log the closing of the election? Not a peep, they just go on and open another election."*
— *"Jolio"*

"You call that an audit log? Everybody's [logged in as] 'admin.'"
— "Lucille Goldman"

"More damning ... is that there doesn't seem to be a document detailing policies and procedures for security both at the user/institutional level and the hardware/software level. There needs to be a document detailing who is entitled to do what with the system."
— "Topper"

"The thing that disturbs me is the comment saying 'add this after it get backs from certification' (or however it's worded). While it's not necessarily nefarious doings — it could be they modified a function, and the mod was crashing, so they didn't want to insert the update it was 'stable' — the note does imply that there may be a non-certified build in use."
— "OutofTouch"

Of course, anonymous participants on an Internet message board are of no help at all if you want to document problems in a formal way. We know very little about these people's expertise or their credentials.

Among the advantages of this informal review format was the perception of protected freedom of speech, moderated to remove obviously disrup-

tive or libelous posts. The DU voting system discussions contained much postulating, backtracking, debating and sometimes plain old ignorance.

Internet forums differ from each other in character. The crowd at Democratic Underground includes many intellectuals, who like to step in to straighten out misinformation, and sometimes get quite fussy about insisting on sources for information posted. Even providing a source doesn't always suffice; a debate sometimes follows about the credibility of that source.

This public "open-source investigation" had many drawbacks, but it did attract intellectual talent and ultimately led to the first formal evaluations of the software outside the voting industry itself. One of the contributors explains how he came to be concerned about the Diebold software:

> "I'm the poor schmuck who configures brand new, untested, computer systems designed by teams of highly educated hardware engineers and loads brand new untested software designed by highly educated teams of software engineers and then performs the 'debug' to make them work together. The systems rarely, if ever, work the first time. It's been my job to be the final arbiter of the finger pointing battles between the two engineering groups who each claim the others product is at fault.
>
> "In short, I have to know enough about the hardware and the software to conclusively prove where the problem lies and then justify pulling overworked engineers off their new assignments to go back and fix something that was considered a 'done deal' under a closed out budget. Not an easy job."
>
> "...In order to survive, programmers tend to be extremely logical thinkers. They exhibit that logical thinking in the way they write their comments into the source code. Each section of code produced by a 'good' programmer has a 'plain english' explanation of what that section does. You might call it a 'professional courtesy' to other programmers who have to work with their code downstream. It's [looking at the comments] a shortcut that quickly lets you know where to focus your attention rather than study every line of code to find what you're looking for. That same logical attitude also drives them to 'ask questions' in their comments when they're asked to do something that's 'illogical' or perhaps they don't understand!
>
> "When you find comments [in the source code] that say things like [paraphrased to take the heat off of list moderators]:

'this is baloney, you don't have to do this, this function is already built in to XXXXXXX, just use the XXXXXX command'

or

'the (insert critical flag here) flag is broken so I did this and that to get around it'

and even things like

'I don't know why you want me to do this, it will let this and that happen....unless that's what you want to happen then I guess it's OK'!

"Comments of this type naturally lead a good programmer looking for problems to investigate what is going on in those routines.

Election systems are 'mission critical' in keeping the full force and power of the United States from falling into the wrong hands. The kind of crap in this code would make it, IMHO, unfit for even checking my e-mail."

— "GoodyTwoShoes"

Another contributor, known here under the screen name "Rummage," studied computer science under a Nobel laureate at Carnegie-Mellon University. In real life and under his normal name, he designs databases for critical applications in the medical field.

"So far, that's the story of the last few days... From databases with no foreign keys (read no referential integrity), unprotected transmission code, ample opportunity for buffer overruns right to PCMCIA slots for wireless modems. Not so much nefarious code as a system with so much opportunity for hacking/fraud as to invite cheating. "

"...as for structure and understanding the DB [database], there are no relationships and the Primary keys are not defined as Access Primary keys. This will make reconstructing the schema a little harder. I don't think a DBA [database analyst] designed this.

"No referential integrity — no autonumber primary keys... Bad for maintaining a reliable database — good for adding and deleting data at will."

— "Rummage"

With the Internet, you never really know whom you are dealing with; a fellow who joins a singles forum may think he's chatting up a buxom blonde named Inga from Denmark while he's actually charming a 400-pound farmer from Iowa named Ralph. I've spoken to many of the participants of the voting machine examination who seemed especially insightful, and they often have impressive credentials, but to most of the world they are anonymous so you can't really know. These informal forum discussions are more akin to casual conversation in the cafeteria than to academic research.

People outside the U.S. are keenly interested in these voting systems. Companies like ES&S and Diebold are marketing their products all over the globe, and some participants in the voting machine discussion confided to me that they are interested in U.S. elections because choices we make directly affect the rest of the world. Here are comments from a European participant who concurs with "Rummage" about weaknesses in the Diebold database design.

> "The fact that they're using Access disallows relationality ... When using a decent database, SQL Server Sybase etc, for example, constraints, triggers, stored procedures, packages, relationships, views, etc are all maintained inside the database — that's where all the business logic resides in a well crafted modern application.
>
> "With Access, however, you're dealing with basically a toy database, and since all of the above are missing, it is common to join tables on the fly using the data connection and SQL code embedded into the program itself...
>
> "... On another note, in a database system, since the system that's updating the database must write the logs, the user in this contect *(sic)* must have write capability to the log table. I could be wrong, but in Access, if you have write capability, you have delete capability...the security features are very limited.
>
> "Security is not something I would consider claiming to have for *any* Access-based application since about any user can gain access fairly easily ... and if you'd ever tried to upsize from Access you wouldn't be touting it as a good thing. Data types get changed, boolean fields don't translate, etc.
>
> "...Sorry, it's a useful tool for basic tasks but compared to a proper database, it's a toy. And it certainly shouldn't be used in a mission critical voting application."
>
> — "t_device"

On forums, people are free to make opinionated, dogmatic and sometimes mistaken statements, just as we do in casual conversation on the subway or in a bar. The Internet culture uses forums and message boards to consider perspectives and ideas, but never for a definitive answer. One reason: It all depends who's chatting that day.

> *"Dear 't_device' — Let's not get into a pissing match. My upsized applications run very nicely to this day. Yes, it's not perfect, but I've used ERwin for documentation and Access is much easier for smaller projects. You get the application running, produce the relational schema and put it on the server. You may choose to develop on the target system. I prefer my method. I hope we can treat each other respectfully.*
>
> *— "Lucille Goldman"*

"I believe we have been civil. If that's not the case, let me know. Apparently we have a difference of opinion. That's healthy. I have upsized a few Access apps and I've developed in it, so I'm not speaking off the top of my head ... Anyway, let's drop the Access better/worse convo and stick to the voting application."

— "t_device"

> "Go over to slashdot [Slashdot.org, a forum for computer people] and try talking about 'security' and 'Access' in the same breath and see how seriously they take you over there — they won't even dignify you with a response, they'll just laugh at you and spray you with onomatopoetic responses like this:
> =====
> slashdot comment:
> *choke*
> *wheeze*
> bwahahahahahahahahahahah
> *gasp*
> Wait, these things are already in use?!?
> *thud*
> ==========
>
> ...because all programmers know there is no security in Access."
>
> — "abcxyz"

> *If you want to know why Access is a bad idea.... ....just do a Google search for 'Access, vulnerability' and browse through the 951,000 hits!*
>
> *— "GoodyTwoShoes"*

Now THAT is a legitimate beef re Access… And the lack of referential integrity (which could have been done, but wasn't) only fuels my suspicions.

— "Rummage"

> *Good point about database audit log tables ... ... very easy to delete any entries. Though there should be some sort of audit ID (in any good database design) that records the sequence of audit log entries which would indicate that a log entry had been deleted.*
>
> *— "gandalf"*

Ahh, the audit log. The more people looked at it, the greater their surprise at the emphasis put on the audit log (by Diebold and its supporters) as a primary security device.

From Dr. Brit Williams*: "Overall security of any computer-based system is obtained by a combination of three factors working in concert with each other: "First, the computer system must provide audit data that is sufficient to track the sequence of events that occur on the system and,to the extent possible, identify the person(s) that initiated the events."[2]

"Generated entries on the audit log cannot be terminated or interfered with by program control or by human intervention."[3] Not quite. This statement is taken from the Diebold document used to sell its system to the state of Georgia, and it refers to a touch-screen audit trail. The server at the county that tabulates all the incoming votes (GEMS) is perhaps a more powerful tampering target, and altering the critically important GEMS audit log is quite easy.

> "Bev, in what way is it significant that the audit log can be rewritten? I'm puzzled by that, because as several people said (I among them) early on, physical control of a machine always means you can overwrite whatever you like. The trick is to keep the bad guys from gaining physical control."
>
> — "Mae West"

---

*Georgia's certification expert from Kennesaw State University.

"The significance is that in letters from certifiers and in documentation provided to certifiers and to the public, they took the curious position that the 'audit log' was a primary means of security protection."

— "BevHarris"

> "Hmmm…did they say in what way? Because if they said it as you implied here (i.e., the existence of an audit file is enough), that would actually be hilariously funny if it weren't so serious. Nerds the world 'round would be cleaning their keyboards and monitors after failing to laugh and swallow at the same time."
>
> — "Mae West"

Looking at the Microsoft Access database used in the main vote tabulation system at the county led to concerns about its audit log and the integrity of the GEMS program as a whole. Interest in the GEMS program began to take on a life of its own on the forums.

"Here's the best part... With GEMS (server) installed on my computer, I was able to create a user name ("me") with a password of my choosing ("mac") and assign myself ADMIN capabilities. This was without ever signing into GEMS....all I had to do was create a new database and I was in like Flynn.

— "BlueMac"

Another forum member pointed out that a database maintenance application might provide the security that GEMS appears to lack.

> "The votes end up in a database whenever there's a database, it makes sense that there would be a database maintenance application. Always preferable to have such an application controlling data entry, to control access and make sure everything agrees, catch entry errors, log activity, etc.
>
> "Without this data entry procedure, what would stop someone from going directly into the database and committing fraud that way? I think you said before that it's an Access database? So open up the database with Access and put your phony votes in. So what I'm saying is the mere ability to edit votes isn't all

> that menacing to me, because it doesn't say that there are no procedures to prevent it from being abused. Maybe elsewhere in the system, or maybe completely outside the system. "
>
> — "Ovaltina"

The GEMS program at the county, which pulls in all the polling place votes, would not be quite as vulnerable if a report was run directly from the voting machines themselves before any data was sent to the county tabulator. This report would have to be run *before* vote tallies begin and posted publicly at the polling place, so that chain of custody of the report does not become an issue. That way, if someone tampered with the central counting (even if they also tampered with the incoming data from the polling place), a red flag would pop up because numbers wouldn't match. Another forum member weighed in:

> "1. Full precinct reports are required by California state law as well as others. The Diebold system better be complying with requirement ...
>
> "2. There is no other auditing function in life that is similar to voting. Once the vote is cast, the identity of the owner of that vote is lost forever! In every other transaction described in these boards, the owner of the data is tied throughout the process. That is why banks can correct your account.
>
> "CA Code 19370 States... At the close of polls... at the precinct...One copy of the statement of return of votes cast for each machine shall be posted upon the outside wall of the precinct for all to see. "The return of votes includes each candidate's name and their vote totals at the precinct. During certification of voting machines, the Voting Systems Panels requires evidence that the procedures of each vendor include this process... "
>
> — "DanglingChad"

Well that makes me feel better. If someone tried to hack the GEMS program, promptly posted reports at each precinct in California (as long as they were printed before any upload of data) would make fraud at the central tabu-

lation stage significantly more difficult, though a clever insider could perhaps get around this excellent safeguard.

Unfortunately, as you'll learn in the next chapter, this procedure apparently was not followed in the 2003 California gubernatorial recall.

Why is it so important to have these printouts done before any data is transmitted to the county? A number of attack points open up while vote data is transmitted to the county by modem. Data transmission does not have to be one-way; the vote tallies might be intercepted, with a revised set zapped back into the polling-place machine during the transmission. Another attack method would be to intercept the vote data with a system that masquerades as the county tabulator, the risk of which is greater if the county transmits votes by wireless methods.

According to the Diebold memos, votes are sometimes transmitted with cell phones, opening up a host of security problems; interception by a spoofed GEMS tabulator is just one risk factor. Any time remote access is gained, the possibility for sending vote data the wrong direction (i.e., replacing the polling place data) arises.

> [In the AccuVote TSx Technical Data Package] "They also make reference to the precinct results being 'reconciled' with the results generated by GEMS at the county office. That's a nice warm fuzzy, if it's Gertrude the election worker taking the printout from the precinct and comparing it with the printout from the county. Unfortunately, I found this reference in a section that refers to the specifics of how the results are modemed in, and it is in a section that specifically deals with communications and the order in which they are transmitted. If the 'reconciliation' is done while this electronic transfer is taking place that's not too warm or fuzzy, is it."
>
> — "BevHarris"

Using the freedom of the Internet, intelligent, concerned citizens began to flesh out issues surrounding electronic voting systems for the first time, using a real system, the Diebold Election System, as their model. I say "for the first time" because until June 2003, only voting-industry insiders were allowed to look at the kind of information these citizens were discussing.

Most of us are given some amount of common sense (as long as sex or money isn't involved), and when we meet up in a group and bring our experiences into the picture, we can make some good, solid decisions. At DemocraticUnderground.com, people familiar with accounting and bookkeeping began to weigh in, and they sometimes took software engineers to task for their failure to understand basic accounting principles.

At issue in this conversation were statements by computer scientists that it was sometimes permissable to design tabulation systems in which totals could be manually overwritten.

> "Each and every vote should exist as a distinct and unadulterated record of one citizen's transaction, probably one or more copies should be generated simultaneously, and everything should be 'journalled' ...
>
> "Since voters are not allowed to recast votes, no possible set of circumstances can possibly exist to justify changing those records.
>
> "... Every change, every addition or subtraction to votes, has absolutely got to be a separate transaction. As a matter of fact, what reason should ever exist to make a change that has an intrinsic value of more than one?
>
> "If a fifty vote change has to be made, then you had better show fifty transactions ... If you need to cancel fifty votes, then you had better show which fifty votes that you are cancelling. Damn and double damn. There is absolutely no technical reason in the world why this cannot be done.
>
> "One vote today is the same as one vote in 1776, which is the same as one vote in 1876, which is the same as one vote in 1976, which should be the same as one vote in 2076.
>
> "What is so hard to understand about that for these computer geeks? "
>
> — "ItAllAddsUp"

A set of User Manual instructions caught my attention. In the GEMS User Manual we found a discussion of how touch screens handle the statistics for undervotes. That's fine, I suppose, but what was it doing in the instructions for how to do manual vote entries?

> "If you have an accounting document, and you are entering the revenues brought in from selling chocolate bars, you don't explain, 'by the way, the correct numbers for the salami sticks you sold should be calculated like this...'
>
> "An entry like that in the chocolate bars accounting instructions would make me go look at what the heck they are doing with the salami sticks.
>
> "By definition, doing manual entry means you are using some form of manual data. It is irrelevant to explain how a touch screen enters votes in a section describing manual entry for manual data. Irrelevant, and also inappropriate. You do not tell people to tinker with the math to make the check sum add up. This is the second such reference — if you'll recall, in the GEMS manual it talks about doing little "adjustments" to the math during manual entry to make sure the check sum is correct.
>
> "...Again, voting is accounting. The procedure they identify is exactly parallel to telling someone how to fudge an accounting log."
>
> — "BevHarris"

"Accounting practices are double entry, not only because of mistakes, but also fraud. Two sources are better than one. So there should be an accounting trail to verify results, especially when there is a question of accuracy ... It doesn't have to be paper but it should be a traceable source document."

— "Cleaver"

Most of all, citizens weighed in with demands for transparency. They chafed at corporate claims to privacy for votes that belong to all of us:

> Bottom line: Government has no business hiding behind proprietary computer code in proprietary voting machines. If the government wants us to use a number 2 lead pencil to mark the ballot, then we damn well better be able to examine that number 2 lead pencil ourselves. We should be able to buy a box of those very same, identical, number 2 lead pencils if we so desire. The paper used for the ballots has got to be paper that can be examined by any who wish. The boxes where the ballots are stuffed need to be made of commonly available

wood, nails, screws, hinges, etc. The boxes need to be able to be examined for false bottoms, hidden slots, etc."

— "ItAllAddsUp"

"States like Georgia have written provisions into their laws that make it impossible to get a machine in dispute adequately inspected. The Georgia law stipulates that three people, a patent attorney and two mechanics, be appointed by law to look at the *computerized* machines! This is tantamount to appointing two blind men and an attack dog to inspect the machine. If either of the 'mechanics' asks about how the machine works the attorney is there to tell them 'it's proprietary information', you're not allowed to know!"

— "GoodyTwoShoes"

Every now and then someone still pops up to tell us that the voting system topic has no legs, or that people just don't care about it. Then explain this: Voting system discussions at DemocraticUnderground.com became kind of an attraction. More and more people tuned in, but at the same time, the subject matter became increasingly technical, while the tone of discussions reflected more urgent concerns. Occasionally someone would sigh and raise their hand:

"Can anyone explain what is happening here in simple language for those of us who are non-techies? I can't make heads or tails about what you may have found here."

— "SkiBob"

Well, we're talking about the computer systems used to count our votes.

"But have you guys found anything? Everybody seems to be talking in very excited tones using terms I can't understand."

— "SkiBob"

(Sorry). Yes, people were finding things. Many of the things they found were eventually found also by researchers at Johns Hopkins and Rice universities,[4] in a report that ended up in *The New York Times*. It was the "increasingly excited tones," in fact, that directly led to the events that produced that report.

> "Attn: BevHarris... look at the cryptographic routines of the voting system. I've just started to go through this system and have a few little snide remarks to make..."

"Topper" was concerned about the possible use of a free, open-source cryptography program which is no longer supported.

> "The problem with using open source with no support is getting a timely answer to your question. Ergo, if there is a security problem during an election, you are stuck with fixing it—which you may not be able to do yourself in a timely fashion."
>
> — "Topper"

"Actually it's not so bad. I'm a programmer and have used that code before. It isn't very well documented and the code is very confusing due to some funky overuse of C++ templates.

"Some of the encrpytion modules are protected by patents which makes it less useful for me but it does appear to be based on an honest attempt to make an open source cryptography library available to everyone for no charge.

"However, I would have to agree that any kind of election software encryption should be based on a standard commercial or government supported encryption solution rather than someone's hobby encryption project."

— "MidniteMunchies"

> "I'm not sure any of the encryption is actually used anywhere ... Since you brought it up, I thought I'd see what algorithm they ended up using. The problem is, I've grepped all over the files, and I don't find any header file inclusions from the crypto library anywhere OTHER than the crypto library. I can't see where the other CVS modules call any of this stuff at all.
>
> "BTW: the library, while perfectly fine for free open source stuff (and I'm an OpenSSL user myself), is a remarkable mish-mash of acquired code. The rijndael.cpp is copied and pasted from the original rijndael.c reference implementation code, there is code copied and pasted from a textbook (dmac.h), the

idea.cpp code is again copied and pasted from the reference implementation idea.c, etc... Not bad for free, but....(apparently not live code anymore either).

"You know, they COULD have gone with OpenSSL — it's free, and supported by far, far more users (and corporate users, such as Apple and IBM for example). But, then again, it doesn't look like they are using any of it anyway..."

— "PoodieToot"

"Mystery solved...but...oh, no... I found what they are actually doing for encryption. They have their own implementation of DES in Des.h

"Here's the bad news...it looks like the DES encryption key is HARD CODED AS A MACRO!!!!!

"AAAAIIIIIIIEEEEEEEHHHHHHHH!!!!!!!!!!!!

"I'll leave discovery of aforementioned key as an exercise for the reader... Good God.....

— "PoodieToot"

"Ooorah!!!!!!! Yeah, I've found the DES.h file...and will start trolling through this...

"If you've hard coded your key and left it just like the public implementation, then it would not be that hard for a hacker to figure out how to get into your system."

— "Topper"

"It would end up as a static string in the executable file And you can tear the static strings out of an executable to view them faster than you can blink your eyes."

— "PoodieToot"

"In your best 50s announcer voice... ... now THAT'S real data security! (cough, cough)"

— "Romeo"

"These things actually use PCMCIA Cards? Huge potential security breaches! Think of the new stuff out there. This is Windows CE based code. Couldn't the existence of these drivers open up any one of these machines having a PCMCIA

based wireless network card installed surreptitiously, allowing remote access via airwaves?

"They're using simple PCMCIA ATA disks These things are basically notepad PC's and the security is almost non-existent. How many local governments will be up on the sophistication required to implement WEP with encryption and hiding SSID's for wireless networks? Heck, you wouldn't even have to hack the wireless network to get around these things, all that is necessary is to pop out one hard drive of results and pop in another with new results preconfigured."

— "Clark Kent"

"Wireless programming required? Are they nuts? i thought I'd been following all the "electronic voting machine" strategies but that's one I missed. I'm a techie, 36 years in the business, some of it with reading punch card votes and optical votes. Wireless programming capability is just plain nuts. That's a security hole the size of a 747.

"That would mean somebody could walk near the voting area (even outside the building), connect to the voting machines via wireless network, and make changes to the voting programs and/or the vote counts"

— "Razmataz"

"I think we've found a potential hole where somebody could alter results remotely with nothing going over any wire. Somebody needs to seriously wardrive elections sites using these things."

— "Clark Kent"

"Ah... That is serious bad news if they are running these terminals wirelessly and only relying on WEP for security. That is enough to fail a security audit at any fortune 1000 company.

"On the other hand, wireless can be extremely secure, more secure in fact that most wired communication if done properly and with the right equipment and design.

"To do it securely, would require fairly recent (and proprietary) technology..certainly not anything that is anywhere near 5 years old."

— "RescueRanger"

"You are assuming no encryption. Because this is wireless does NOT mean no encryption is being used. WEP anyone? Proprietary encryption perhaps? But then again it could be none is... "

— "spock"

"The onus is on the local election administrators, though I have my home wireless network locked down so tight most wardrivers will take one look at all of my security measures and drive on down the street to the guy who is advertising an SSID that is the default on the access point he installed and has never changed the admin password.

"Even I know that with 128 bit encryption using WEP, no advertised SSID, and a MAC Address list can still be cracked. MAC addresses can be spoofed relatively easily and brute force can break the 128 bit encryption if you've got the processor power. Even with encryption, it can be cracked. Now tell me how many of the local election boards you've had experience with are sophisticated enough to implement WEP, let alone MAC Address access lists etc. etc. etc.?

"Add to that the fact that there is a ton of code that could hold back door access and this thing is rife with potential abuse.

"Nope, this doesn't even compare to the potential for pushing out chads on hundreds of cards with a pin so they register as double votes and thus are spoiled ballots. The potential for abuse is magnitudes above this. If the government does not require an independent code review by at least three different companies, it's not doing its job."

— "Clark Kent"

"I trust you are aware... The chances of breaking 128 bit encryption with a brute force approach could very well take centuries with just about any computer on the planet?"

— "spock"

"A 128 bit encrypted file and the encryption level on WEP are two different things. I assure you, WEP is crackable. A PGP file with 128 bit encryption is, as you stated, not easily crackable. And when database files have passwords that are the name of the county where votes are counted, how secure is this system?

— "Clark Kent"

"Perhaps this programmer's comment in the Results Transfer Dialog file [TransferResultDlg.cpp] will answer that question for you: 'Changed the election.dbd file to only store ascii code not unicode to make it compatible between windowsNT/95/98 and WinCE. The convertion from acsii to unicode, if required, is done when the data is retrieved from the database. Note: This does not affect rtf data since it is always stored in ascii.'"

— "BlueMac"

"STRAIGHT ASCII???????? For compatibility with Windows 95/98/NT???? On February 15, 2001?????"

— "Clark Kent"

"Why not? ;o "

— "spock"*

"That's some encryption there! Straight ASCII for backwards compatibility on operating systems that are obsolete. This makes a lot of sense for a system we are supposed to trust the future of the world to."

— "Clark Kent"

"I believe it is talking about the unencrypted values for backwards compatibility when being viewed. But then again that's another problem with leaked source that may or may not be final, you can't be sure."

— "spock"

"And that's the problem with computer voting systems, isn't it... You can't be sure."

— "PoodieToot"

---

* ;o is a keyboard code meaning "wink"

"If I were the guys doing openssl, I'd be real pissed off right now. That blows chunks. I guess assigning a public/private key pair to each networked voting machine is too difficult for the people entrusted with the lifeblood of democracy?"

— "mortal"

"Seems a Congressional investigation should be next."

— "SPacific"

But a congressional investigation was not what came next, or even after next, or even next after next after next. If anything should have a congressional investigation in full view of TV cameras, the voting industry should, but as of the writing of this book, it hasn't happened.

What came next was a quiet phone call on a Sunday morning.

* * * * *

Over the course of a year, I consulted with about two dozen computer techs. Several are not on Democratic Underground because they are Republicans. I met one on Free Republic, a conservative forum. One was a former client of mine. Voting system integrity is a truly nonpartisan subject — Democrats, Republicans, Libertarians, and Greens — everyone but the Charlatan Party, I guess — all respond the same way when someone says, *By the way, we will no longer be auditing the vote, thank you*.

Among my sources is a computer programmer I'll call "Cape Cod."

The best programmers explain things in a very concise way. I'm stubborn enough that I'll keep asking until I understand the answer or the other person starts shouting at me, whichever comes first. But highly skilled programmers are extremely organized thinkers, and it is easy to follow their explanations. "Cape Cod" is such a person. His explanations of complex computer concepts follow this simple, linear fashion: *Here is A, and I'm going to take you to B. Take hold of A, and walk just this way, and I'll describe the scenery as we go. Now, here we have arrived at B; did you enjoy it?*

"Cape Cod" rarely calls me and has always been irritatingly discreet about his examinations of the Diebold files. When he calls, his clipped, East Coast voice provides no unnecessary words and gives very tidy explanations. He also never

calls unless he has something to say. He made one efficient, four-minute call to explain how a voting system might be able to cheat with 'zero reports,' for example:

"It's quite simple, really; your goal is to stuff the electronic ballot box while at the same time generating a report at the beginning of the election which tells you that zero votes have been cast, proving the ballot box has *not* been stuffed.

"Here's what you do: You stuff the ballot box by entering two vote totals that cancel each other out: 'plus 50 for Truman, minus 50 for Dewey.' You have thus created a spread of 100 votes between the candidates before the election begins — yet because +50 and -50 sum to zero, you have added no extra voters.

"To make the report read zero when you start the election, simply instruct the code to put a string of zeroes into the 'zero report' if there are any negative numbers in the ballot-stuffing area, but it must only do this if there are no other votes in the system. And by designing a database without referential integrity, you can arrange for the evidence of this ballot-stuffing area to fall off the radar."

(Did you understand that? I did — and he only had to explain it once.)

One Sunday morning while I was still in my bathrobe, I received one of "Cape Cod's" rare phone calls.

"Go to your computer. I want to show you something."

He proceeded to walk me through the process of rigging an election using a real Diebold program, with a version used in a real election, with a vote database for Cobb County, Georgia, found on the Diebold Web site.

Quick overview of GEMS: The GEMS voting software collects votes from the polling places, tabulates them and generates reports. GEMS is used for both optical scan ballots (where you fill in a dot, or draw a line to your choice) and touch-screen machines.

After the polls close, poll workers transmit the votes that have been accumulated to the county office. They do this by modem or by taking out the memory card (like a disk, but the size and shape of a credit card) and driving it over to the county office.

At the county office, there is a "host computer" (also called the "server"), which has the GEMS program on it. It receives the incoming votes and stores them in a vote ledger.

Bypassing the Supervisor Password

The GEMS User Manual tells us that the default password in a new installation is "GEMSUSER." If you install GEMS, click "new" and make a test election, then close it and open the same file in Microsoft Access, you will find an encrypted password in the "Operator" table. Anyone can copy an encrypted password from there, go to an election database and paste it into that using Microsoft Access. Using this method you can open any election database with the password "GEMSUSER."

You can grant yourself supervisor privileges by making yourself an "admin." You can add as many friends as you want. (I added 50 of mine and gave them all the same password, which was "password.")

Using this simple way to bypass password security, an intruder or an insider can enter the GEMS programs. However, you don't even need a password to go in the back door.

The GEMS program looks and feels very secure when you work with it. Running behind the GEMS program is a database using Microsoft Access. When you open an election in GEMS, it places an election database in a folder on your computer. Anyone who has Microsoft Access on their computer can open this election file, simply by double-clicking the file, going in the back door. This kind of access is not certified or authorized, but it can be done anyway.

If someone gains access to GEMS by getting at the computer in the county office, or by hacking in through the Internet or a phone line, they can get hold of this election file.

Back to "Cape Cod."

"Here's what we're going to do," he said. "We'll go in and run a totals report, so you can see what the election supervisor sees. Then I'll show you something unusual."

I opened the GEMS program and ran a totals report. Then I ran a detail report showing the results in each polling place.

"Now, open the file in Microsoft Access."

"Close out of GEMS?"

"No, Access is configured for multiple users."

OK, I didn't know that. Two people can wander around in the vote database at the same time without bumping into each other.

Remember that there are two programs: the GEMS program, which the election supervisor sees, and the Microsoft Access database (the back door) that stores the votes, which she cannot see.

When you open the election database in Microsoft Access, you will see that each candidate has an assigned number. One of the tables tells you the number for each candidate. You can then click a table called CandidateCounter, which will show you how many votes the candidate has accumulated for each polling place.

On this day, "Cape Cod" showed me another table in the Cobb County file, called SumCandidateCounter. This table had the same information as the first, but we observed that it had two complete sets of the same information. One set was marked by a flag, the number "-1."

Notice that this gives us three sets of votes.

"Change some of the vote totals in SumCandidateCounter."

I did, choosing votes from the set that did not have the "-1" flag.

"Now let's run a report again. Go into GEMS and run the totals report."

The totals report showed my new numbers, proving I could alter the report by going in the back door and replacing vote totals with my own in the unflagged votes in the SumCandidateCounter table.

"Now go back and look at that detail report."

The detail report had the original votes, not the ones I changed. It was drawing its information from either the CandidateCounter table or the flagged set in SumCandidateCounter. In accounting, this is called having two sets of books. (Or in this case, three. I never heard what the third set of books does. "Cape Cod" called it the "Lord only knows" table.)

"Why would it be good to have the detail report show the real votes while the summary shows the ones I changed?"

"This allows the system to pass a spot check."

Does this modification produce an audit trail?

Not if you go in the back door while the supervisor has the election open.

Any time you open the GEMS program, it will show up in the GEMS audit log. But suppose you want to erase yourself?

In the Diebold system, it seems that everyone uses the same name when they go into GEMS (they all call themselves "admin"), but I wanted to see whether I could become someone new, play around in GEMS and then erase myself from the audit log.

I created a new user by the name of "Evildoer." Evildoer performed various functions, including running reports to check his vote-rigging work, but only some of his activities showed up on the audit log. For some reason, a few of his activities omitted themselves from the audit log even before I tampered with it. But I wanted to erase *all* evidence that Evildoer had existed.

I went in the back door by double clicking the GEMS database on a computer with Microsoft Access loaded on it. I expected the audit log entries to be numbered automatically with something I could not edit. That way, if I erased some Evildoer activities, the numbers would still be there, marking an activity that had disappeared. I was surprised to find that I could just type new numbers over any of the GEMS audit log numbers, and I could also erase events altogether.

In every version of GEMS that I examined, the autonumbering feature was disabled, allowing anyone to add, change and delete items from the audit without leaving a trace. Soon, there was no trace of Evildoer in the audit log.

Going back into GEMS, I ran an audit report to see if Evildoer had indeed disappeared. As Verbal Kint, in the movie *The Usual Suspects* (1995) said, "The greatest trick the devil ever pulled was convincing the world he didn't exist."

Another thing that seemed improper in the GEMS program is this: You can enter *negative* votes. It is a simple matter to program the software so that it will never accept a negative number. Why should it? A vote total that is less than zero can only be illicit.

The entire process — bypassing the password, changing the vote totals, cleaning up the audit log — took less than 10 minutes.

* * * * *

During the month of June, I hadn't seen much of *Scoop Media*. But Scoop's publisher, Alastair Thompson, is never far from a phone when he smells something breaking.

"Hi, Bev. (New Zealand pronunciation, "Bivv"). Alastair here. (New Zealand pronunciation "Alasteh"). What's up?"

"Well, we have a pretty important story. With the GEMS program, using one of the databases found on the FTP site, we were able to rig it," I said.

"Hmm!"

"I'm writing it up. I'm not sure where I'm taking it, though."

"You know, I rather thought this might be a good time to publish the link," said Thompson.

—*Come again???* "What link?"

"Oh you know. To the files."

"The files from the FTP site?"

"It seems like a good time, don't you think? I think we should come out with your story at the same time. Get people to it, right?"

"To the link."

"Right."

"Alastair, that set of files is huge. Do you have the bandwidth?"

"Oh, I think we'll be all right.They have bandwidth to burn."

The story went out on *Scoop Media* on July 8;[5] Thompson ran one story about the hackability of GEMS, along with another editorial which he titled "Bigger than Watergate!" He has since been roundly criticized for that choice of title, but remember: Watergate took two years to get as "big as Watergate."

Just sixteen days after Thompson posted the article that brought the world to the link, *The New York Times* posted a scathing report on the Diebold voting system software, by computer security experts from Johns Hopkins and Rice University who had downloaded the files from *Scoop Media*. At least one new story came out in a major media outlet every day for the next two months. In

September, a report written by Pentagon contractor Scientific Applications International Corp. (SAIC) was published that detailed 328 security flaws in the Diebold voting system, 26 of which it deemed "critical."

Stories have now begun to surface about conflict of interest at the top of the company and secret lobbying efforts. People are starting to follow the money trail behind the voting machines.

"Bigger than Watergate." Ha!

Perhaps 50 years from now, some intrepid reporter in a far-flung corner of the world will be scoffed at for titling his article "Bigger than Votergate."

* * * * *

On July 24, 2003 *The New York Times*[6] ran an exclusive story about "stunning, stunning security flaws" uncovered by four researchers at Johns Hopkins and Rice universities. The report, titled "Analysis of an Electronic Voting System" described many of the same findings as those pointed out by the irreverent bunch at Democratic Underground. It was blistering. The Hopkins/Rice report quoted source code explaining its weaknesses, and delved into Diebold's smart card security and its source code architecture and provided the first detailed critique of Diebold's failure to use cryptography correctly. The report also revealed that one of the flaws had been pointed out by voting examiners five years ago and still had not been corrected.

Diebold Election Systems came out swinging:

• The software was never used in any election!

• Well it was used in some elections, another Diebold spokesman was reported to have said, by *WiredNews* reporter Louise Witt.[7] I called her to ask how solid this quote was. Rock solid, she said, but the quote was pulled a day later in favor of this: A small part of the software may have been used in some elections.

• The software is old and out of date, Diebold decided. An article in *The Plain Dealer* [8] pointed out that Diebold was preparing to sell Ohio its new TSx system, though the company admitted it might not be certified by purchase time.

Most of the people I've interviewed about this say the software cannot have been rewritten and tested in the short time since July 24 — or even in the 10

months since the last election. The problems exist in the program itself and patching them will not produce a sound voting system.

Nevertheless we are told by Diebold that the problems:

1) Are fixed

2) Were never a problem in the first place, because the Diebold software is surrounded by election procedures and physical security, which have effectively neutralized the problems all along.

There are weaknesses in the Hopkins/Rice report. Several sections seem to assume that touch-screen machines are connected to the Internet; nothing I've seen indicates that is the case. I have seen indications that the GEMS servers connect to the Internet, and GEMS also connects to a digiboard which, in turn, connects back to touch-screens with a modem when the election closes.

Before the election, GEMS loads ballots into the touch screens, but everything I've seen indicates that this is done using touch screens placed in the office near the GEMS machine, rather than loading the ballots over the Internet.

The criticism that the Hopkins/Rice report doesn't take into account all the election procedures is, in many ways, absolutely correct. It doesn't appear that the authors read the user manuals that go with the software; they apparently did not interview any election officials, either. Several of the concerns in their report prove unfounded when you find out more about election procedures.

Other areas of the report describe cracks that would be impractical or could not affect many votes at a time. The most publicized security flaw in the report has to do with making extra voter cards (or reprogramming one so that it can vote as many times as you want). These are valid concerns, but checking the number of voters signed in against the number of votes cast is a required safeguard in most states and would quickly reveal such a ploy. This type of hack would also be very difficult to achieve on a grand scale; you would have to make rigged smart cards and send people in to cast extra votes at hundreds of polling places at once, which gets into the crazy conspiracy realm.

The biggest taint applied to the Hopkins/Rice report is a conflict of interest on the part of one of its primary authors, Aviel Rubin.

Lynn Landes, a freelance reporter, revealed that Rubin had been an advisory-board member for VoteHere, a company that claims its software solves many of the problems in the Hopkins/Rice report.[9] Rubin also held stock options in VoteHere; he resigned and gave back his stock options, but not until after Landes published her article. Rubin told Landes that he had forgotten about this conflict of interest when he wrote the report.

Three more researchers — Dan Wallach, who is a full professor at Rice University and Adam Stubblefield and Yoshi Kohno, of Johns Hopkins — also wrote the report, and none of them appear to have any conflicts of interest. It seems unlikely that all three would help Rubin slant a report just to help him sell VoteHere software.

The importance of the Hopkins/Rice report:

1) It correctly identifies weaknesses in Diebold's software development process. The code is cobbled together to fix and patch. The correct way to produce quality software is to first develop a precise schema (structure) that says what the software must do; the development process must test against this schema to see that it performs flawlessly. Instead, Diebold's software engineers seem to make it up as they go, and this is evident both in the source code and in their internal memos.

2) It identifies very real security flaws that can jeopardize vote data, especially during transmission to the county tabulator.

3) The Hopkins/Rice report pushed media coverage into the mainstream. And because, when you are researching this story, you can't even sneeze without finding something new, coverage of the integrity of our voting system will continue to gather momentum. The longest leap forward in a single day was due to the Hopkins/Rice report.

4) The report triggered another evaluation, this time by the SAIC.

SAIC report

In August 2003, the governor of Maryland, which had recently placed a $55 million order for Diebold touch-screen machines, ordered an evaluation by Scientific Applications International Corp.[10] There are concerns with this report as well

— though the report is 200 pages long, two-thirds of it was redacted. In the small part that was made public, more sections were redacted, including everything about GEMS except a general statement that it was unsatisfactory.

If Rubin is said to have a conflict of interest, then SAIC has a whopper: The vice chairman of the SAIC, Admiral Bill Owens, is the chairman of VoteHere. Like the Rubin report, the SAIC report identifies many areas that VoteHere claims to have the solution for.

The SAIC report validates important findings in the Hopkins/Rice report and identifies many new areas of concern. Because it is heavily redacted, we don't know the details on all of the flaws it found, and many are specific to Maryland. Still, these words reverberate since Diebold's software is still being used in elections:

The system, as implemented in policy, procedure, and technology, is at high risk of compromise. Application of the listed mitigations will reduce the risk to the system. Any computerized voting system implemented using the present set of policies and procedures would require these same mitigations.

or to put it more succinctly. "328 security flaws, 26 deemed critical."

Chapter 10 footnotes

1 – Digital Millenium Copyright Act of 1998 http://www.loc.gov/copyright/legislation/dmca.pdf

2 – "Security in the Georgia Voting System," April 23, 2003, by Britain J. Williams, Ph. D.

3 – Georgia RFP Sales Proposal for Diebold Election Systems: Phase I, Tech Proposal.

4 – Analysis of an Electronic Voting System, Johns Hopkins Information Security Institute Technical Report TR-2003-19, July 23, 2003. http://avirubin.com/vote/

5 – *Scoop Media*, 8 Jul 2003; "Sludge Report #154: Bigger than Watergate" http://www.scoop.co.nz/mason/stories/HL0307/S00064.htm

6 – *New York Times* 24 July, 2003; "Computer Voting Is Open to Easy Fraud, Experts Say" http://query.nytimes.com/gst/abstract.html?res=F70A15F73E5B0C778EDDAE0894DB404482

7 – *WiredNews.com* 4 August, 2003; "More Calls to Vet Voting Machines" http://www.wired.com/news/politics/0,1283,59874,00.html

8 – *Cleveland Plain Dealer* 14 August, 2003; "Voting machines under review in Columbus" http://www.ohiocitizen.org/moneypolitics/2003/voting.htm

9 – *EcoTalk.org* 18 August, 2003; "Voting Machine Fiasco: SAIC, VoteHere and Diebold" http://www.ecotalk.org/VoteHereSAIC.htm

10 – "Risk Assessment Report Diebold AccuVote-TS Voting System and Processes" 2 September, 2003; Science Applications International Corp.