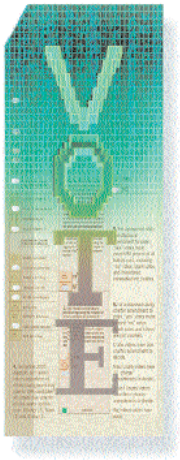
AUDITING

BY **DOUGLAS W. JONES**

ELECTIONS ARE CONCEPTUALLY TRIVIAL: After everyone has voted, the votes in each ballot position are counted, and the winner of each race is declared. Barring the use of some form of ranked-preference ballot, the algorithms required are hardly more than mere summations. What makes elections difficult is the problem of performing this summation over a canvassing hierarchy spanning many polling places and local election offices, where every single participant has a vested interest in the outcome. Complete trust cannot be extended to any single authority to run an honest election because every candidate for such trust may have a vested interest and may end up abusing that trust. Therefore, voting systems must be secured not only against improper actions by voters and election officials, but also against improper actions by programmers, technicians, and system administrators.



The integrity of our elections is guarded by using two broad classes of defenses: The first involves an array of preventive measures; obvious examples include the requirement that voters identify themselves and the requirement that ballot boxes be locked and sealed during voting. The second line of defense involves auditing measures that detect error or fraud and, in the best case, allow reconstruction of the correct election totals despite these events.

Canvassing an election is an accounting function, where votes, not currency, are the subject of the count. Consequently, election auditing resembles financial auditing. As with financial auditing, an audit can be initiated in response to suspicion of impropriety, but auditing is at its most effective as a deterrent to fraud and error if it is also conducted routinely.

One key requirement of governmental elections vastly complicates the job of auditing: the requirement for a secret ballot. The details of this requirement vary from jurisdiction to jurisdiction and are weakened by the use of absentee ballots, but generally, voters are forbidden from retaining anything that could be used as proof of how they voted. This prevents the use of classic auditing tools such as the issuance of receipts, and severely curtails what may be retained in transaction logs.

In some jurisdictions, ballot secrecy is not absolute: In Great Britain, for example, the Secret Ballot Act of 1872 requires that each ballot cast be tied to the voter who cast it by records that are held as a state secret [10]. Where such rules apply, auditing is easier than in jurisdictions where the right to a secret ballot is absolute, as in the U.S. In Iowa, for example, it is illegal for an election official to mark a ballot in a way that allows the voter to be identified [9].

ELECTIONS

Exploiting the redundancy in election records to conduct useful audits and improve the system design process.

Even when ballot secrecy is not absolute, approaches to auditing elections that do not involve access to complete records remain valuable. Routine audits that require the invasion of voter privacy are bad policy because they weaken the voter's confidence in the secret ballot.

Effective auditing of elections raises several important issues. What do we do if a routine audit discloses an error? Do we invalidate the election? Do we initiate a complete recount? The answers to these questions are outside the scope of this article, but it is important to note that appropriate answers to these questions are needed before many policymakers will support effective auditing measures.

Auditing

The central requirement for a system to be auditable is that it retain sufficient information to allow detection and correction of error and falsification. In this regard, auditable systems share many characteristics with fault-tolerant systems.

For a system to be auditable, however, we add an additional requirement that follows from our interest in deterring malicious attacks. We require that the redundant elements be carried in the custody of independent actors, to the extent that independence can be assured, and we insist on preservation of evidence, so we rely on indelible or write-once media. Thus, we issue carbon copies of the paper receipt for a financial transaction to both parties in the transaction, and we develop systems such as double-entry bookkeeping.

Auditing offers no guarantees, however: collusion between the partners in a transaction allows the issue of dishonest receipts, and there is always the possibility that the collusion will involve the auditors themselves. For particularly important transactions, we therefore ask for additional witnesses.

In the context of elections, auditability has sometimes been referred to as universal verifiability: "The election results should be verifiable to independent observers (or any interested party, for that matter). This means it is possible to check unambiguously that the published election result corresponds to the ballots cast by legitimate voters" [11].

Redundancy in Election Records

A basic auditing measure that can be applied to any voting technology has been advocated for some time: the maintenance of a record, outside the voting machine, of the turnout, or the number of ballots that should have been issued. As of Election 2000 in the U.S., there were still 12 states that did not require reporting of the turnout, while in many other states, these numbers come out long after the election [8]. One measure of the turnout is a count of the signatures in the poll-book, but in jurisdictions where voters sign serial-numbered affidavits of

eligibility on entry to the polling place, the simplest turnout measure is the difference between the first and last affidavit number issued at each polling place.

If we carry this turnout figure forward through the canvassing process, we can check, at each level, to see how many votes remain to be accounted for. Because the number of ballots actually counted may differ from the reported turnout, we must introduce a new figure into our accounting—the number of ballots unaccounted for (see Figure 1).

The proper measure of turnout is more complex than suggested here due to problems introduced by provisional ballots. Provisional ballots are included in the number of affidavits of eligibility, where that system is used, but they are excluded where poll-books are used. Postal voting adds additional complexity; an absentee ballot request serves as affidavit of eligibility, but because of postal delays and other losses, the number of ballot envelopes received should also be counted.

$$S + P + X = B + F$$

S: Signatures in poll-book
P: Provisional ballots
F: Fleeing voters who did not vote
B: Ballots found in ballot box
X: Ballots improperly stuffed into ballot box

All quantities are non-negative
Non-zero F and X are indications of trouble

Figure 1. Figures to carry forward and reconcile at each canvassing level.

We can perform a similar check within each race on the ballot, adding the number of votes for each candidate to the number of abstentions in that race and the number of invalid votes (for example, overvotes). This sum should equal the number of ballots counted; where it does not, the difference provides a measure of the error in the count; see Figure 2. (In elections where voters are entitled to cast more than one vote and in various forms of ranked-preference voting, more complex rules can be formulated to accomplish this goal.)

$$B = C + O + U$$

B: Ballots found in ballot box
C: Sum of votes for specific candidates
O: Number of overvotes
U: Number of undervotes

All quantities are non-negative

Figure 2. Figures to reconcile for each race at each level in the canvass.

Exploiting Redundancy

The redundancy described here can be used to detect both error and fraud. If we perform these checks at every level in the canvassing hierarchy, we will detect most clerical errors and we will defend against the simplest forms of ballot-box stuffing. These checks are at their most powerful if we transmit the turnout figure up the canvassing hierarchy through an independent channel from that used for the actual ballots or vote subtotals. For example, if we maintain and transmit the turnout figures using manual methods outside the computerized election system, we can conduct an end-to-end check on the entire computerized election system. California law comes close to requiring this [3].

This illustrates a common aspect of many auditing measures. While they may be added to a system for auditing purposes, their most important use is in immediate self-auditing. Their most important use is not in some rare retrospective analysis of an election, but rather, to detect errors before the canvass of votes is completed, so that the errors may be investigated and corrected before any figures are released to the public.

These measures can be compared to backward error correction; like checksums in conventional data transmission, they allow us to determine when retransmission is required, but we would have to include additional redundant data to allow reconstruction of corrupt data.

An important characteristic of these measures that distinguishes them from backward error correction tools such as cyclic redundancy checks or block checksums is that they are simple computations that can be done directly by people, including not only election officials but also outside observers. This is important in contexts where the correctness or honesty of the machinery itself is subject to question.

These measures are generally sufficient to detect any single error, so they will easily detect and allow correction of the kinds of clerical or computer errors that plague all complex distributed processes. What these measures cannot detect is the improbable accident that adds votes to one candidate in a race while subtracting votes from another. Unfortunately, this is exactly the result that would be expected from carefully constructed corrupt voting software or insider manipulation of the data.

Auditing After the Fact

The preceding measures apply to the data itself, but they do not address the authenticity of the data. Authenticity may be checked by software using cryptographic techniques, but a human auditor needs simpler means, particularly when assessing the record of an election after the fact.

One important aspect to examine is the chain of custody for each piece of evidence pertaining to the election. What machinery produced this data, who collected it from the machine, and how was it pre-

served? What we need is analogous to the documentation for the chain of custody required to bring evidence to court in a criminal case. Some of this can be maintained automatically. For example, voting machines have long maintained records of the number of ballots cast. Electronic voting machines generally record, in their audit log, records of the time the machine was turned on or off, the time it was opened and closed for voting, and the time each ballot was cast [6]. In order to preserve ballot secrecy, the time records for the ballots must not be tied to the individual ballots, but are maintained separately.

Given such records, an auditor can compare the number of vote-cast audit records with the number of ballots found in the ballot box. Furthermore, if the serial numbers of the machines at each polling place are recorded in the handwritten records of the polling place, these can be compared with the serial numbers recorded electronically, both in the audit logs of the machines and in the files representing electronic ballot boxes.

These are not sham audits, as some have charged. These checks can quickly determine that the votes from some machines were not counted in the canvass, and they can detect the substitution of the electronic record from one machine for the electronic record from another.

Design for Auditability

The audit rules suggested here do not detect the shifting of votes from one candidate to another within one election, so we must erect additional barriers. We would like to be able to recount the election itself, but in order to reduce the likelihood that we will need to do so, we can incorporate firewalls into the design of our voting system that stand in the way of transfers of votes from one candidate to another in the same election.

How we elect to raise these firewalls is secondary to this discussion; clearly, strong protection mechanisms, type-safe languages, and cryptographic tools can all play important roles. As one example, we could adopt the rule that votes and vote totals for each candidate are encrypted using different keys, and that the software that manipulates or transmits this data never makes use of more than one key at a time.

Inspecting software to assure that internal firewalls isolate key components from each other is far simpler than inspecting it to determine if more general correctness criteria are met. Such firewalls can be established relatively easily except at the input interface; in today's voting systems, this will typically be either a touch-screen based graphical user interface or an optical mark-sense scanning device. The software at this interface must be subject to far more intense scrutiny.

Unfortunately, in several of today's voting systems, the software at this interface is legally exempt from scrutiny under exemptions written into the voting system standards in order to encourage use of industry standard third-party components [7]. While well intentioned, these exemptions completely ignore the possibility of Trojan horse attacks from within such components.

Where we can identify difficult-to-audit components, we can attempt to cut them off from access to critical data using more internal firewalls. For example, we know that many Trojan horse attacks are time dependent (including the original), so we can attempt to prevent access to the system's time-of-day clock and real-time clock from all system components except those with an absolute need for these services.

The successful creation of such barriers within a computer system requires that all system interfaces be known. As a general rule, operating systems that contain proprietary or hidden interfaces should not be used in secure systems, since each such interface could be the basis of a covert channel allowing system components to tunnel through the firewalls that we need to erect.

Recounts

If our election system retains the original ballots, a genuine recount is possible. This is always possible if original paper ballots have been preserved, but it may also be possible with some forms of electronic records. What matters, from an auditor's point of view, is that this record indisputably captures the voter's intent, and that these records can be examined without the use of any of the software components that may be subject to question in the audit.

As already noted, indisputable capture of voter intent by a fully electronic system is difficult because of the difficulty of auditing the user interface component of the voting system. One solution to this problem is to turn the auditing job over to the voter, printing out a copy of the ballot captured from that voter and inviting the voter to verify this copy. If used purely for auditing and recount, this markedly strengthens the audit. If this voter-verified document is used as the legal ballot (for example, by using it as input to an optical mark-sense scanner for tabulation) then, between the auditing performed by the voters themselves and occasional recounts, we have achieved the goal of end-to-end auditability in elections.

Hand recounts have been subject to controversy because of doubts about the ability of people to perform an accurate count, but by avoiding use of any mechanism and by being open to public observation,

they eliminate all questions about the honesty of the mechanisms used. This is why California law has long required hand recounts of randomly selected precincts after every election [4].

Where hand recounts cannot be done, reasonable auditing considerations dictate that the recount be conducted using a different mechanism and different software than was used for the first count, as required in Arizona [2]. Unfortunately, there are several states where the law requires the exact opposite, asking that all recounts be done on the same tabulator using the same software as was used on the first count [5]. This makes it difficult to ask for a recount that tests for errors in the mechanism or software, seriously weakening the whole idea.

Parallel Testing

When a complete end-to-end audit is not possible, for example with purely electronic voting systems, there is another option, parallel testing. This involves pulling randomly selected voting systems from service and testing them by entering fictitious votes and comparing the results from the machine with observations of the inputs that would be illegal with real votes.

The challenge posed by parallel testing is that of assuring the voting system cannot detect that it is being tested and not used in a real election. Ideally, the system to be tested should be selected as the polls are opened, so that no pre-election setup procedure could inform it that it is under test, and the test should be performed for the full period the polls are open, with a number of voters and an arrival distribution typical of the election itself. Furthermore, the voting system should be unable to detect any monitoring mechanisms attached for the purpose of this testing.

Parallel testing can disclose some problems that are very difficult to find in other ways. For example, it can disclose improper ballot presentation. Conventional auditing, focusing on the records of the votes cast, cannot detect that a choice, or indeed, an entire race, was not offered on an electronic display screen, or that alternatives or races were presented in the incorrect order.

Conclusion

Several ideas mentioned in this article bear repeating. First, auditing is at its most effective when the computations required to perform an audit are sufficiently simple that many observers can perform independent audits. This is one of the central aspects of systems that rely on a voter-verified paper trail, where each voter has an opportunity to help audit the one system component that is most difficult to audit without this features.

Second, if an understanding of the audit requirements is developed early in the design of a system, this can drive the design process, leading to compartmentalization requirements that, in turn, simplify the conditions that must be proven in any verification effort.

We have left significant questions unanswered here. Primary among these is the question of what to do if an audit detects problems in an election. We have also ignored questions of sample size; clearly, statistical sampling methods are applicable to any partial audit of an election, and they also apply to parallel testing. This has been addressed elsewhere [1].

Finally, it is important to note that auditing cannot detect all problems. It cannot detect violations of voter privacy or incorrect ballot presentation. Testing—in particular, parallel testing—is necessary to defend against these problems. C

References

1. Adler, J. Confidence: What it is and how to achieve it. *NIST Symposium on Building Trust and Confidence in Voting Systems*. December 2003, Gaithersburg, MD; www.votehere.net/papers/NIST_121003.pdf.
2. *Arizona Revised Statutes, 2003*. Title 16, Chapter 5, Article 12, Section 16-664—Recount of votes by automatic tabulating system; www.azleg.state.az.us/ArizonaRevisedStatutes.asp.
3. *California Elections Code, 2003*. Chapter 4—Official canvass, Section 15302—General provisions; www.leginfo.ca.gov/calaw.html.
4. *California Elections Code, 2003*. Chapter 4—Official canvass, Section 15360—One percent manual tally; www.leginfo.ca.gov/calaw.html.
5. *Colorado Revised Statutes, 2003*. Title 1, Article 10.5, Section 108—Method of recount; www.state.co.us/gov_dir/leg_dir/geninfo.htm.
6. Federal Election Commission. *2002 Voting System Standards, Volume I*. Section 2.2.5, System audit; fecweb1.fec.gov/pages/vss/vss.html.
7. Federal Election Commission. *2002 Voting System Standards, Volume II*. Section 5.2, Basis of software testing; fecweb1.fec.gov/pages/vss/vss.html.
8. Hargrove, T. Voter totals often not known. *Cincinatti Post*. (Dec. 12, 2000); www.cincypost.com/news/2000/undsid120500.html.
9. *Iowa Code, 2003 Supplement*. Section 39A.3, Election misconduct in the second degree; www.legis.state.ia.us/IACODE/2003SUPPLEMENT/39A/3.html.
10. *Procedures at a General Election*. Department for Constitutional Affairs, London, August 2001; www.dca.gov.uk/elections/ge2001/procedures/.
11. Schoenmakers, B. Fully auditable electronic secret-ballot elections. *Xootic*. July 2000; www.win.tue.nl/xootic/magazine/jul-2000/schoenmakers.pdf.

Douglas W. Jones (jones@cs.uiowa.edu) is an associate professor in the Department of Computer Science at the University of Iowa and has served for a decade on the Iowa Board of Examiners for Voting Machines and Electronic Voting Systems.