Understanding the Security Properties of Ballot-Based Verification Techniques (Short Paper)

Eric Rescorla
RTFM, Inc.
ekr@rtfm.com

Abstract

As interest in the concept of verifiable elections has increased, so has interest in a variety of ballot-oriented mechanisms that offer the potential for more efficient verification than traditional precinct- or machine-level audits. Unfortunately, threat analysis of these methods has lagged their design and in some cases implementation. This makes it difficult for policy makers to assess the merits and applicability of these techniques. This paper provides a fairly non-technical description of the security threats facing these systems with the intent of informing deployment decisions.

1 Introduction

The ultimate objective of a transparent, verifiable, election is to allow any observer to convince himself that the reported vote tallies — and in particular the reported winners — were correct. In current elections, this role is filled, when at all, by *precinct-based* or *machine-based* audits: subtotals are published for each batch of ballots; batches are randomly selected for audit and then hand counted; the hand count subtotals are compared to the reported subtotals. This procedure does provide for third-party verifiable elections, but at a significant cost in time and effort, especially for close races.

A number of authors [15, 6, 17] have suggested the use of *ballot-based auditing*, in which the batch size is a single ballot. Ballot-based audits offer much higher statistical power and thus the potential for a higher level of verifiability with far less effort. While I am unaware of any ballot-based audits which have yet been conducted, recently, there has been significant interest in an even more radical approach, exemplified by the Humboldt Election Transparency Project (ETP)[1]. In the ETP, ballots are rescanned and the images are published on the Internet and via DVDs. Third parties can then independently (and remotely) process the images and compare the results to the reported precinct totals.

Most of the published work on ballot-based approaches has focused on the mechanics of operating the audit, with only a limited amount of attention paid to threat analysis. While that analysis is relatively straightforward and well understood in the security community, it has not been explicitly stated in a form suitable for general readers, but rather is mostly implicit in work such as [6, 22]. This paper represents an attempt at a self-contained threat analysis targeted for readers outside the computer and voting security community. Our focus is on optical scan (opscan) systems because they seem to be easier to secure and because the records they produce are more suitable for auditing than those of DREs, which are susceptible to presentation attacks like those described by Everett [11].

2 Background: Third-Party Verifiability

Before discussing the security of various auditing strategies, it's important to get a clear picture of what we're trying to accomplish. Words like "transparency" and "verifiability" get used fairly loosely in the context of elections, but if we're going to try to enforce security properties, we need to be concrete about what we're trying to achieve.

Figure 1 shows an abstract model of a typical optical scan election; diagrams like this appear all over the voting literature and so this should be fairly familiar.

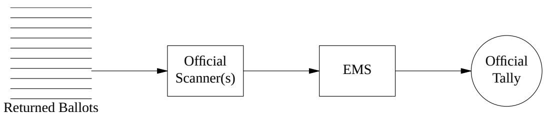

Figure 1: Abstract election model

[1] http://humtp.com/

The input to the counting and tabulation system is one or more sets of completed paper ballots filled out by voters (possibly using some sort of ballot marking device). These ballots may come from multiple sources, such as different precincts, absentee ballots, vote-by-mail, etc. The ballots are then fed in batches into one or more optical scanners that read them. The scanners may either be in precincts or centralized. The scanner output is then fed into the *Election Management System* (EMS), which tallies up the results, determines the winner of each contest, etc. The final results are then published, including both per-batch subtotals and the final totals for each contest. Note that the exact division of labor between the scanner and the EMS varies between different voting systems. In some systems the scanner knows the structure of the ballot and interprets the votes and in other systems the scanner just produces images that are processed by some software application. For example, in the Hart InterCivic system, the precinct-count scanner (eScan) processes the ballots and emits *cast vote records* (CVRs) that just contain the selections on each ballot. By contrast, the Hart central-count system uses a commodity scanner attached to a computer running a special application (BallotNow) that interprets the results and then feeds them into the EMS, that may or may not be running on the same computer as BallotNow.

Our primary security objective is *third-party verifiability*: we would like an observer who is not affiliated with the election officials and has no special access to be able to determine with some level of (statistical) confidence whether the paper ballots match the reported totals. More concretely, if the election was correctly run, we should be able to convince that observer that if they were given access to the ballots and could hand count them themselves, they would get the same results as the official tallies.

When seen from that perspective, Figure 1 is somewhat misleading: the outside observer has no real way of determining what processes the election officials use to count the ballots. Even if he knows what equipment was purchased and watches the counting process, the equipment is effectively a black box: he has no way of knowing that some adversary (whether insider or outsider) has not replaced the software on either the scanner or the EMS with malicious software that records results of his choice. What the observer sees is better represented by Figure 2: ballots come into the system and then some opaque process occurs and results are emitted. In order for the election to be third-party verifiable, an observer needs to be able to check that the results are correct without any visibility into the internals of the scanning or counting process.

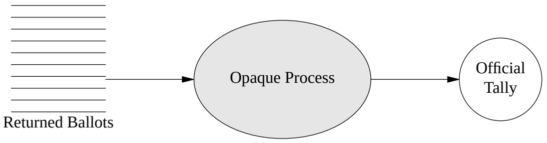

Figure 2: The outside observer's view

There is a direct parallel here to the concept of *software independence* [19]: a software independent system allows the verification that an election was counted correctly without any trust in computer equipment that may have been used in the election. A third-party verifiable system allows a third party to check the correctness of the tabulation of a set of ballots without any trust in (or even knowledge of) the mechanisms used to tabulate those ballots. Another, more paranoid, way to look at this is that we *assume* that the scanning and tabulation is being run by someone who would like to steal the election and the purpose of verifiability is to ensure that they do not. Obviously, in the vast majority of cases election officials are honest, but our intention is to design a system that is secure even if they are not.[2]

Note that in in this section we are implicitly assuming the set of ballots we are using for the audit has not itself been tampered with. For instance, in typical central count optical scan systems the ballots are collected at the precincts and then returned to election central for processing. If some ballots were stolen or replaced in between the polling place and election central, then the eventual count will be incorrect. This form of attack is not detectable by simply recounting the paper ballots. In current systems, protection of the paper ballots themselves is generally accomplished with physical and administrative controls, although it may be possible to use auditing techniques to provide an additional level of security. We discuss the impact of physical security on auditing further in Section 4.

3 Verification Methodologies

In this section we provide an overview of a number of potential mechanisms providing third-party verifiability and discuss their security properties.

3.1 Conventional Precinct-Based Audits

The traditional recommendation is to perform either a machine- or precinct-based audit. In this form of audit, units (either machines or precincts) are selected at random and all the ballots in the unit are counted by hand. The hand counts are then compared with the published

[2] I owe this general formulation of security to Steve Bellovin, who described the communications security threat model as "You give the packets to the attacker to deliver."

subtotals and mismatches can be investigated. If the discrepancies are large enough, the selection and hand counting procedure may be repeated with more ballots until either the result of the election has been confirmed (i.e., the discrepancies are too small to have affected the winner) or the entire election has been recounted by hand. There has been extensive work (see for instance [20, 21]) on the statistics of this form of recount, including how to select units for audit, when escalation is necessary, etc. Once the subtotals are confirmed (or corrected), any third party can add them up themselves to verify that the final totals are correct.

In order for the tabulation process to be third-party verifiable it's critical that the *subtotals be published*[3] before it is known which precincts will be audited. Otherwise, someone wishing to tamper with the election might wait until the precincts were selected for auditing and then attack only those precincts which had not been selected, thus evading the audit. This also means that the audit units must be selected by *verifiably unpredictable* process such as dice rolling [10] or a cryptographic pseudorandom number generator [7].[4] Otherwise, someone looking to commit fraud might be able to control or predict which audit units will be selected. If both of these conditions are met, any third party who observes the ballots being audited can verify that the election results are correct — even if they were not able to observe the original counting process or, due to ordinary human limitations, could only observe parts of it — because the audit provides a verifiable check on the counting process.

The major challenge with audits of this type is that they are extremely expensive. The smaller the margin of victory, the more units must be audited, and if the scanner error rate is at all significant, the audits tend to escalate very quickly. For example, if we want to have a 95% chance of detecting any error that covers 1% or more of the ballots this might require auditing over 25% of all ballots even if no discrepancies are found during the audit, clearly a major effort. Hall [16] reports that the required California 1% manual recount takes nearly the entire 28-day statutory period in Los Angeles County. A 25% audit in such an environment seems likely to be prohibitively expensive. A good technical introduction to precinct-based audits can be found in Hall [12], Aslam, Popa and Rivest [1] and Stark [21, 20].

3.2 Ballot-Based Audits

Because of the high cost of precinct- and machine-based audits, there has recently been significant interest in *ballot-based* audits [15, 6, 17]. The idea with a ballot-based audit is that instead of auditing the results of a single precinct or voting machine, we select individual ballots for auditing and compare the results against the reported results from the scanners, as shown in Figure 3, which was adapted from Sturton, Rescorla, and Wagner [22].

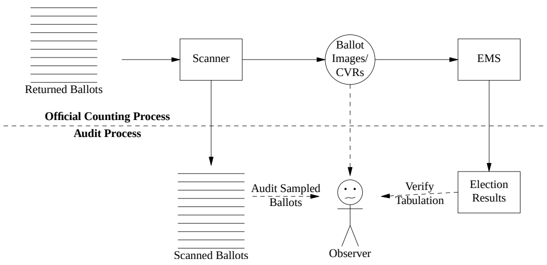

Figure 3: Ballot-based auditing workflow (adapted from Sturton, Rescorla, and Wagner [22])

Much of the workflow in a ballot-based audit is the same as in the ordinary *optical scan* (opscan) setting shown in Figure 1. However, because a ballot-based audit requires comparing individual paper ballots to the corresponding electronic records, the election officials must make electronic records for each ballot available to the auditor, either as images or as *cast vote records* (CVRs): a list of the selections for each contest for each ballot. The audit then consists of comparing randomly selected paper ballots to their electronic counterparts.

As before, any third-party observer can add up the electronic records for themselves. This is most easily done if CVRs are published, since then a spreadsheet, calculator, or even independently written software can be used to do the tabulation. The actual audit of course requires physically handling the original paper ballots to compare them to the electronic records and observers must be able to watch this. As with precinct-based audits, if a ballot-based audit is conducted properly a third-party observer can verify that the election results were correct without any need to trust or even observe the original counting process.

While it might seem like images provide a higher level of verifiability than CVRs because they give the third party more data to work with, it's important to remember that the purpose of the sampling phase of the audit is precisely to verify the scanner's interpretation of the ballots, so having the images does not assist this process. Thus, the audit checks the CVR creation process itself. Indeed, from an ease of use perspective, images rather

[3] Technical note: it's actually sufficient for the subtotals to be committed to prior to the audit. For instance, the county could publish a cryptographic hash of the subtotals. However, it is not clear what advantage this provides.

[4] Technical note: It is important that the observer be able to determine for themselves that the selection was random. Otherwise, it might be possible for insiders running the audit process to avoid selecting ballots where fraud has been committed. Hardware based random number generators are generally not suitable here because it is very difficult for an observer to verify that they are functioning correctly.

than CVRs make the problem significantly worse: absent special tools like the blink comparators used by pre-computer astronomers, it is much harder to compare a scanned image to a physical object than it is to compare it an electronic CVR to a paper ballot. Moreover, if images are used, third parties must first process them and map them onto CVRs before tabulating subtotals and comparing them to the reported subtotals. This introduces another potential source of error: if a ballot is ambiguous and the third-party scanner interprets it differently than the original scanner, this will show up in the subtotal results, but will be fairly difficult to track down because any ballot in the batch could be wrong. By contrast, if CVRs are compared, then this ambiguity has been removed and any errors are serious. In addition, publishing ballot images introduces potential privacy problems (see Section 5). For this reason, publishing CVRs seems superior to publishing images.

The major benefit of a ballot-based audit is that it requires auditing far fewer ballots for any given level of statistical power. To continue the example above, in order to have a 95% chance of detecting a 1% level of fraud would require sampling something less than 300 ballots as opposed to over 100,000 for a precinct-based audit in a county the size of Santa Clara, California (on the order of 500,000 voters). This is a counterintuitive result, but is uncontroversial statistically. Appendix A provides some background that may help the reader get a feel for the statistics.

There are two major challenges for any ballot-based auditing system. The first is having some method for finding a given ballot out of the set of ballots. One could imagine simply manually counting into each stack, perhaps inserting markers at convenient spots or separating the ballots into stacks of a given size (cross-stacking) [18] to facilitate future indexing into the same stack, however this is likely to require manual counting—which is inherently error-prone—of a large fraction of the stacks. Potential optimizations include marking the ballots with a serial number [15, 6] or weighing the ballots [22]. All of these approaches have some drawbacks and as far as I know, none has been used to conduct a ballot-based audit, so it is unclear what the most efficient method is. The second challenge is maintaining voter privacy. We discuss this extensively in Section 5.

3.3 The Humboldt Election Transparency Project

The Humboldt Election Transparency Project (ETP) has some similarities to a ballot-based audit, but also is different in some important respects. Figure 4 shows the model.

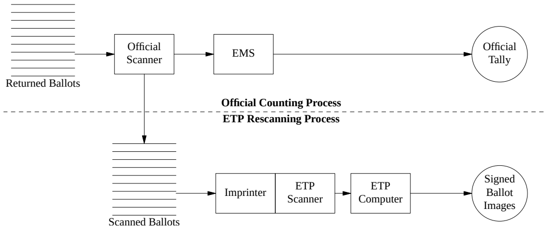

Figure 4: The ETP model

As in the previous section, the top half of the diagram shows the standard opscan model. The bottom half represents the new functions performed by the ETP. Once the ballots been processed by the official scanner, they are then processed by the ETP in sequence through an imprinter built into the scanner and which marks them with a serial number,[5] a commodity optical scanner (Fujitsu 5900c), and then a computer that captures the images as individual JPEG files, packages them as ZIP files, and digitally signs the archives with PGP. The files are then made publicly available along with their digital signatures.[6] Third parties who want to verify the machine count can download the images and process them themselves, either using their own image processing software or with the Ballot Browser software [7] developed by Mitch Trachtenberg for the ETP.

The security properties of an ETP-style system are more complicated to evaluate than those of the previous two systems we have described. The difference hinges on the level of trust placed in the ETP rescanning process: If the imprinter, scanner, and computer used to rescan the ballots are trusted, then in effect the ETP becomes a second, distributed, machine-based recount. Anyone with access to the ballot images (that effectively means access to the Internet), can independently verify that the ballot images correspond to the posted subtotals, which also implicitly checks the interpretation of each image by the scanner, since misinterpretations will affect the final count. This clearly allows the detection of some kind of errors. Indeed, the ETP has already detected a significant bug in the Diebold/Premier GEMS system, where one deck of ballots could be lost [4] (the so-called "zero deck bug").

However, the ETP has a built-in asymmetry: if the third party does *not* trust the ETP systems and personnel, then they can only partially verify the election. In particular, they can only verify that the scanners are not misin-

[5] In California, ballots may not have serial numbers preprinted on them. Other jurisdictions have serialized ballots and so could potentially omit the imprinting step, though it may still be necessary in order to find the ballots again.

[6] `http://earc.berkeley.edu/hosting.php`

[7] `http://www.tevsystems.com/warning.html`

terpreting ballots (by comparing the subtotals to an independent machine interpretation of the ballot images) and that the EMS is correctly resolving the contests based on subtotals (by independently tabulating the subtotals). However, this is insufficient to provide third-party verifiability of the election results: if the ETP systems and/or personnel were malicious, they could substitute false ballot images during the re-scanning process. These images would match the reported subtotals and totals, but would not accurately reflect the paper ballots fed into the scanner. Thus, independent machine interpretations (whether by Ballot Browser or independently developed software) will produce results that match the false images and thus will also match the false subtotals and totals. From the perspective of such an observer, both the original vote totals and the ETP scans are part of the same opaque counting process, just as if there were no ETP. Thus, the system does not meet our objective of being secure even in the face of insider attack.

This *image substitution* attack cannot be detected without direct comparison of the paper ballots to the scanned images. Importantly, the digital signatures applied by ETP do not provide any security against this attack because they are applied *after* the scanning process: if the scanner has been subverted then the digital signature is being applied to bogus data and so does not provide any additional security against this attack. Note, however, that the signatures do prevent an attack in which an attacker distributes bogus copies of the ballots that were not generated by the ETP, so it is an effective defense against some forms of outsider attack.

Although substitution attacks are clearly theoretically possible, it's natural to ask whether an attack of this sort is actually plausible. Trachtenberg [23] argues that it is not and cites several security features:

- The scanner is a commodity product.
- The software run on the attached host is open source.
- The images are digitally signed.
- The images themselves can be matched against the paper record.

The first three features are a significant obstacle to *outsider* attack: it would be difficult to insert the appropriate malware into, for instance, the Linux distribution and if the machines are kept under tight physical control and never networked, it is difficult to see how an outsider could compromise them. Note that the machines must be kept under physical control at all times, not merely during the audit. If the machines are ever left out of control, an outside attacker might be able to install some form of unremovable malware[8] or simply tamper with the hardware. Generally available tamper seals are simply not up to the task of protecting commodity hardware from subversion. Similarly, the digital signature precludes tampering of the images once created, as long as the private key is controlled adequately and the public key is disseminated securely.

However, these features all require a third-party observer to trust election officials and ETP personnel. It would be a simple matter for an insider to load subverted software onto the attached computer or potentially the scanner (internally, most scanners are general purpose computers attached to scanning hardware, and many have replaceable firmware). The substitution attack software could tamper with the scanned images to make them match the votes desired by the attacker. This type of image manipulation is comparatively easy to automate, since it's primarily a matter of cutting and pasting one well-specified part of an image onto another, if not just replacing images wholesale. There is currently no practical way for a third party to verify the software running on a commodity system of this type, thus making this form of attack undetectable to an observer.[9]

This leaves us with the final defense: matching the images against the paper record. This, of course, is a ballot-based auditing process with serialized ballots and would be performed very much like the audit described by Calandrino et al. [6]. Without this step, there is no way for a third-party observer to verify that the counting was conducted correctly. With this step, the ETP becomes effectively a ballot-based audit as described in Section 3.2, except that it publishes images rather than CVRs, which, as we have said, is less convenient.

From a third-party verifiability perspective, then, the rescanning and signing process does not add much value beyond the ballot-based audit itself: The final ballot-based audit serves a check on the entire process, even if the scanner is compromised, so having two scans, neither of which can be trusted, is not really better than one. If the voting equipment would emit CVRs the rescanning process can be safely omitted. However, if the certified voting equipment will not emit CVRs or images and modifying it is problematic, rescanning may be the most efficient way of supplying the raw material for a ballot-based audit. As discussed in the previous section, it would probably be more convenient for the rescanner to emit CVRs rather than images.

[8] Technical Note: his is possible, using for instance, e.g., a BIOS rootkit, which can persist past operating system reinstallation.

[9] Technical note: while there are systems such as Trusted Platform Module (TPM) (`http://www.trustedcomputinggroup.org/developers/`) that are intended to allow the remote verification (attestation) of the software on a computer the number of un-attestable components of this system including the scanner and the connection to it makes that form of defense impractical in this case, even if it were otherwise practical.

4 Tampering With Ballots

In the previous section we focused on the task of trying to reconcile an electronic count with a set of paper ballots. However, if the ballots themselves have been tampered with, then auditing may not be sufficient to detect errors. We need to be concerned with ballot tampering in two places: (1) before the ballots are counted, e.g., in transit from the polling place to election central, and (2) after the ballots are counted, e.g., in storage between the time of counting and the audit.

4.1 Tampering Before Counting

Ballot tampering before the counting process is extremely difficult to detect with any auditing process; auditing relies on reconciling two different kinds of records, but before the ballots are counted the primary records we have are the ballots themselves. While some forms of tampering such as ballot box stuffing or the theft of large number of ballots can be detected by comparing the number of recorded voters with the number of cast ballots, these are of limited value against ballot modification or substitution attacks. Instead, administrative controls (sealed boxes/bags, two-person chain of custody rules, etc.) currently provide the main line of defense.

Because precinct count optical scan systems count ballots immediately after voters cast them, they minimize the risk of tampering before counting at the cost of creating a long post-counting exposure window. In order to exploit that window, an attacker must attack *both* the ballots and the precinct scanner. However, existing scanners have been shown to be relatively easy to attack (see, for instance, [5, 14, 3]) and as the precinct scanners are under the control of poll workers, it is unclear to what extent precinct count provides an independent check on poll worker ballot tampering. In systems where the results of the precinct scanning are independently published (e.g., by posting a summary tape outside of the polling place), precinct scanners do make ballot tampering by non-pollworkers significantly more difficult.

It is also possible to run *both* precinct count and central count optical scan systems, as suggested by Calandrino et al. [6]. The primary advantage of such a system over a pure precinct count system is that it is robust against ballot reordering in between the precinct count and the audit: because ballot based auditing systems require the ability to find individual ballots and ballots are likely to be reordered in transit, it is difficult to do a ballot based audit using precinct count equipment alone. However, if the ballots are scanned at the precinct as well, this serves as a check on ballot tampering en route to election central.

Note that cryptographic “end-to-end” systems (there are many of these, see for instance [9, 2]), provide another form of independent check: because individual voters can track their ballots through the entire counting process, ballot tampering is automatically detected, even without an audit. Unfortunately, such systems are not currently widely deployed, and to date interest in them among voting officials has been marginal at best.

4.2 Tampering After Counting

Tampering can also occur after the counting process. If only the ballots—or only the electronic records—are tampered with, then this creates discrepancies that are readily detectable by the auditing techniques described in Section 3. If the paper ballot is treated as the final record, then paper-only tampering may be sufficient, but it seems likely that a significant discrepancy between the paper and electronic records would trigger an investigation. Thus, tampering only with the paper is only really useful to cast doubt on an election, not to change the reported counts. A more attractive avenue would be to use a paper-based attack to supplement an electronic attack, thus hiding evidence of tampering with the scanner from an audit.

In principle, a third, independently developed, set of records might provide an additional check on an attacker who tampered with *both* the official scanner and the paper ballots. For instance, in the ETP the ballots are scanned twice, once officially and once by ETP personnel. If an attacker were to subsequently tamper with the scanner and the paper records, this could be detected by reference to the ETP output. However, unless ballot handling procedures are significantly weaker after scanning than before, then it is unclear why an attacker would not simply tamper with the ballots prior to scanning, thus evading this form of checking as well as other forms of audit.

Moreover, this check is only third-party verifiable if a third party can convince themselves that the additional records were not tampered with, which is generally not true. In particular, in the case of a system like ETP, the records are generated by an unobservable piece of software and therefore third parties cannot verify that there was not tampering by insiders.[10]

[10] It may be possible to partially restore third-party verifiability by replacing the second scanner with a simpler device that is harder to subvert. For instance, one might feed the ballots into an old model (non-scanning) photocopier or a film camera, in which any software or firmware is not integrated into the reproduction path, thus making any sophisticated ballot replacement/modification impractical. The new records (paper copies or film) could then be kept separately from the ballots, forcing any attacker to tamper with them as well in order to avoid detection. The obvious disadvantage of this sort of mechanism is that comparison of the analog records to the original ballots is difficult.

5 Privacy Issues

The most difficult problem with any verification system that operates on the level of individual ballots is preserving the secrecy of the ballot. In order for third parties to independently verify the process of tabulating the individual ballots, the contents of each ballot must be published somehow. Any time individual ballots are published, there is a risk of having those ballots linked to individual voters. We need to consider both attacks in which the voter's privacy is involuntarily violated and attacks in which a voter cooperates to expose his ballot, as in vote-buying and coercion situations. Unfortunately, only partial solutions are known at this time.

Vote Order The most obvious avenue of attack is vote ordering: if the ballots are published in the order in which voters voted, then an attacker who observed the order in which ballots were dropped into the ballot box/scanner has an opportunity to link votes to voters. This attack does not require any cooperation from voters and could be mounted on a retail basis by pollworkers or election observers. This is obviously most plausible for precinct-based in-person voting and much less plausible for absentee and vote by mail. Sturton, Rescorla, and Wagner [22] suggest explicitly shuffling the ballots prior to scanning, and it may be the case that in some environments that shuffling happens in the ordinary ballot-handling process. Hildebrand [13] suggests printing a random number on each ballot and then sorting the ballots by that random number which is effectively a more deterministic form of shuffling. This is a topic that deserves future study.

Stray and Variant Marks If images rather than CVRs are posted, there are many opportunities to mark a ballot in ways that do not impact processing but yet are distinctive. For instance, stray marks in strategic locations could be used to indicate the voter's identity. If this was done carefully enough, it might even be possible for a pollworker to do it without the voter noticing by marking a corner or some other non-coded region. While it may be possible to remove this sort of marking by electronically or physically masking off everything on the ballot that does not correspond to selections, this opens up the system to complaints that the images have been manipulated prior to being posted, which is precisely what posting images instead of CVRs is intended to counter.

Even if we only consider selection regions, there are many opportunities for distinctive marks. Any examination of real optical scan ballots quickly reveals that voters regularly do not follow directions: They X rather than fill, fail to fill in the regions (ovals, arrows, etc.) completely, overfill the regions, make hesitation marks in other regions, etc. Because these marks go to the interpretation of intent, they cannot be masked off without compromising the utility of the image. It's unclear whether enough information can be encoded here to easily detect voter identity. It may not be possible to deliberately encode the information, but there is probably enough variation to distinguishing one ballot from another. For instance, a voter might make some semi-distinctive marks and then photograph their ballot with a camera phone (often illegal but as a practical matter very difficult to detect) and then allow the person they are attempting to prove their vote to to compare their photo to the scanned images, counting on natural variation as a distinguisher.

Systems that use ballot marking devices are generally less vulnerable to this kind of attack, although it is still potentially possible for a voter to surreptitiously mark their ballot before or after the BMD. However, BMDs are relatively uncommon and transitioning to them would involve a major capital expenditure for many jurisdictions. Moreover, if BMDs are used we now face the problem of attacks by the BMD itself. For instance, the BMD might change the voter's vote and hope the voter didn't notice; Everett's [11] results suggest this is likely to be effective. While this is not a threat to verifiability as we have framed it, is a threat to the security of the election.

Write-Ins Write-in ballots clearly have the potential to allow easy signaling of voter identity. This is most easily dealt with by replacing the actual ballot with a record that it contained a write-in without specifying who the vote was for. All the write-ins for a given race can then be treated as a single pseudo-candidate[11] and if they are large enough to affect the race (e.g., the difference between the least popular winner and most popular loser) can be hand-counted separately. If images are published, it is probably easiest to simply not publish images for write-in ballots (though they still must be available for audit sampling), replacing them with a dummy ballot indicating the presence of a write-in. An alternate option would be to (very carefully) mask out the write-in field.

Pattern Voting Even if only CVRs and no write-ins are published, we still need to be concerned with *pattern voting* attacks. In such an attack, the vote buyer wishes to influence a few contests and is indifferent to the rest. He instructs the voter to vote for the candidate of his choice in the relevant contest and then to vote in a specific pattern (designed to not be a pattern likely to be selected by an ordinary voter) for the rest of the contests. He then checks the published ballots for that pattern.

[11] This has similarities to Stark's [20] pooling strategy for losing candidates.

The feasibility of a pattern voting attack depends on the complexity of the ballot compared to the size of the pool within which a voter must be distinguished. For instance, if there are 10 separate contests with even two candidates per contest this suffices to encode around 1,000 separate identities, even if we avoid choosing some voting patterns that will naturally occur.[12] This is not a particularly complicated ballot: the Santa Clara County ballot for the 2008 general election had over twenty contests with many having more than two candidates. (Remember that undervotes and overvotes can be used to signal information here as well). If ballots are labelled by precinct (or distinguishable because of the set of contests) and a precinct is around 300-1,000 people, then this level of encoding is enough to distinguish individual voters.

In principle, one could turn to having a separate paper ballot for each contest or small number of contests, but this would require major changes to election procedures. While it is possible to cryptographically disaggregate the votes, as far as I know the available methods are relatively complex and difficult for ordinary voters to understand. This inherent tension between vote secrecy and third-party verifiability is a major challenge for any ballot-based auditing scheme.

6 Summary

Dissatisfaction with traditional precinct-based audits has spurred new interest in alternative, individual ballot-oriented verification techniques. These techniques offer the possibility of equivalent or even superior levels of verifiability with far lower levels of manual effort. Unfortunately, far less public attention has been paid to the threat analysis for these techniques, which is substantially different than that for traditional audits. We have compared the security models for two such techniques, ballot-based auditing and the Humboldt Election Transparency Project to traditional precinct-based audits.

When an ETP-style approach is used without any observable comparison of the paper ballots to the ballot images, it can provide a check on voting machine error. However, without that comparison it is not third-party verifiable. With such a random comparison, it, like ballot-based audits, provides a third-party verifiable check on the entire election. The rescanning procedure used by the ETP may be able to detect some kinds of errors and, may provide a form of backup or partial substitute for the physical security of ballots after scanning, but from a third-party verifiability perspective it does not add any security value above that provided by a ballot-based audit. However, it may be necessary for use with voting machines which do not output either ballot images or CVRs. Both systems offer inferior privacy guarantees to that provided by precinct-based audits. In general, the publication of ballot images has worse privacy than the publication of CVRs.

Policy makers considering moving towards ballot-based verification techniques should carefully consider the threat environment and evaluate what balance of features and risks best suits their deployment scenario.

Acknowledgements

Thanks to Richard Barnes, Steve Checkoway, Joe Hildebrand, Hovav Shacham, Terence Spies, and Cynthia Taylor. for their helpful comments on this paper. Thanks also to the anonymous EVT reviewers and our shepherd, Joseph Lorenzo Hall.

References

[1] Aslam, J. A., Popa, R. A., and Rivest, R. L. On auditing elections when precincts have different sizes. *USENIX/ACCURATE Electronic Voting Technology Workshop 2008* (July 2008). http://www.usenix.org/events/evt08/tech/full_papers/aslam/aslam.pdf.

[2] Benaloh, J. Simple Verifiable Elections, Aug. 2006.

[3] Blaze, M., Cordero, A., Engle, S., Karlof, C., Sastry, N., Sherr, M., Stegers, T., and Yee, K.-P. Source code review of the Sequoia voting system. Part of [8], Aug. 2007.

[4] Bowen, D. California Secretary of State Debra Bowen's Report to the Election Assistance Commission Concerning Errors and Deficiencies in Diebold/Premier DEMS Version 1.18.19. http://www.sos.ca.gov/elections/voting_systems/sos-humboldt-report-to-eac-03-02-09.pdf, March 2009.

[5] Calandrino, J. A., Feldman, A. J., Halderman, J. A., Wagner, D., Yu, H., and Zeller, W. P. Source code review of the Diebold voting system. Part of [8], Aug. 2007.

[6] Calandrino, J. A., Halderman, J. A., and Felten, E. W. Machine-assisted election auditing. *USENIX/ACCURATE Electronic Voting Technology Workshop 2007* (Aug. 2007). http://www.usenix.org/events/evt07/tech/full_papers/calandrino/calandrino.pdf.

[7] Calandrino, J. A., Halderman, J. A., and Felten, E. W. In defense of pseudorandom sample selection. *USENIX/ACCURATE Electronic Voting Technology Workshop 2008* (July 2008). http://www.usenix.org/event/evt08/tech/full_papers/calandrino/calandrino.pdf.

[12] Technical note: with two candidates per contest, there are $2^{10} = 1024$ possible patterns, ($3^{10} = 59049$ if we allow undervotes; $4^{10} = 1048576$ if we allow both under and overvotes). However, some of these patterns, e.g., straight party line voting, will be commonly used by voters who simply have that preference, so the number of patterns which act as a useful signal is something less than you would otherwise expect. An anonymous reviewer suggested to me that ordinary voters might randomize their down-ticket votes to mask such patterns, but it seems doubtful that this would be common enough to have a significant impact on the ability to verify the "correct" vote.

[8] California Secretary of State D. Bowen. "Top-To-Bottom" Review of voting machines certified for use in California, 2007. Online: http://sos.ca.gov/elections/elections_vsr.htm.

[9] Chaum, D., arback, R., Clark, J., Essex, A., Popuveniuc, S., Rivest, R. L., Ryan, P. Y., Shen, E., and Sherman, A. T. Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes, July 2008.

[10] Cordero, A., Wagner, D., and Dill, D. The role of dice in election audits—extended abstract. *IAVoSS Workshop on Trustworthy Elections 2006 (WOTE 2006)* (June 2006). http://www.cs.berkeley.edu/~daw/papers/dice-wote06.pdf.

[11] Everett, S. P. *The Usability of Electronic Voting Machines and How Votes Can Be Changed Without Detection*. PhD thesis, Rice University, 2007.

[12] Hall, J. L. A quick primer on the mathematics of post-election audit confidence, Mar. 2007. http://www.josephhall.org/eamath/eamath.pdf.

[13] Hildebrand, J. Personal communication.

[14] Inguva, S., Rescorla, E., Shacham, H., and Wallach, D. Source code review of the Hart InterCivic voting system. Part of [8], Aug. 2007.

[15] Johnson, K. C. Election Certification by Statistical Audit of Voter-Verified Paper Ballots, October 2004. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=640943.

[16] Joseph Lorenzo Hall. *Policy Mechanisms for Increasing Transparency in Electronic Voting*. PhD thesis, University of California, Berkeley, 2008. http://josephhall.org/papers/jhall-phd.pdf.

[17] Neff, C. A. Election confidence—a comparison of methodologies and their relative effectiveness at achieving it (revision 6), Dec. 2003. http://web.archive.org/web/20060117190359/http://www.votehere.net/papers/ElectionConfidence.pdf.

[18] Norden, L., Burstein, A., Hall, J. L., and Chen, M. Post-election audits: Restoring trust in elections. Brennan Center for Justice at The New York University School of Law and The Samuelson Law, Technology and Public Policy Clinic at the University of California, Berkeley School of Law (Boalt Hall), Aug. 2007. http://www.brennancenter.org/dynamic/subpages/download_file_50227.pdf.

[19] Rivest, R. L., and Wack, J. On the notion of "software independence" in voting systems, July 2006. http://vote.nist.gov/SI-in-voting.pdf.

[20] Stark, P. B. Conservative statistical post-election audits. *The Annals of Applied Statistics 2* (July 2008), 550–581.

[21] Stark, P. B. CAST: canvass audits by sampling and testing. Tech. rep., University of California at Berkeley Department of Statistics, Feb. 2009. http://statistics.berkeley.edu/~stark/Preprints/cast09.pdf.

[22] Sturton, C., Rescorla, E., and Wagner, D. Weight, Weight, Don't Tell Me: Using Scales to Select Ballots for Auditing. In *Proceedings of EVT/WOTE 2009* (Aug. 2009), J. L. Hall, D. Jefferson, and T. Moran, Eds.

[23] Trachtenberg, M. Can't Digital Images Be Faked. http://democracycounts.blogspot.com/2008/07/cant-digital-images-be-faked.html, August 2008.

A A Brief Overview of Auditing Statistics

Many people find the mathematics of auditing counter-intuitive. This Appendix provides a brief, non-rigorous, overview of the main facts about auditing, and in particular why ballot-based auditing is more efficient than precinct-based auditing.

Imagine a very simple election consisting of a single contest with only two candidates, Alice and Bob and conducted on optical scan equipment. We have 1,000,000 voters organized into 1,000 separate precincts with 1,000 voters per precinct. The preliminary vote tally is recorded as 510,000 votes for Alice and 490,000 votes for Bob. The object of the audit is to convince ourselves that if we were to hand recount the entire election Alice would still be the winner. In order for such a recount not to show Alice as the final winner, at least 10,000 ballots must have been misread by the scanner: if 10,000 votes for Bob were transferred to Alice, that would change a 500,000-500,000 tie into a 20,000 vote win for Alice. Because each precinct has 1,000 voters in it, that means that this fraud would require that at least 10 precincts (1%) had paper ballots that don't match the scanner results. Let's call those precincts "bad" and the others "good" and assume that all the bad ballots are isolated into 10 bad precincts.

A Precinct-Based Audit Let's assume that indeed 1% of precincts were tampered with and ask how likely it is that a precinct-based audit will detect the fraud. We start by picking a single precinct at random and count it by hand. If it doesn't match the original count we've found a bad precinct. This is evidence of error and we can start an investigation if necessary. However, if it matches, we could have just gotten lucky and picked a good precinct (remember that the vast majority of the precincts are good: The chance that the first precinct we select will be good is 990/1000). Now, if we select another precinct at random out of the remaining precincts, the chances that it will be good is 989/999 (remember we've already audited one precinct and found it good.) So, the chance that the election was tampered with and we find two good precincts at random is these two numbers multiplied together:

$$\frac{990}{1000} \times \frac{989}{999} = 0.98009 \qquad (1)$$

Thus, we have a 2% chance of detecting the fraud by auditing two precincts. Auditing three precincts gives:

$$\frac{990}{1000} \times \frac{989}{999} \times \frac{988}{998} = 0.97027 \qquad (2)$$

In other words we have approximately a 3% chance of detecting the fraud. We can keep adding new precincts in exactly this manner and while I'll spare you the math, if we want to have a 99% chance of detecting the fraud, we need to audit 368 distinct precincts, or over a third of the votes. This is quite expensive. Note that if we do detect errors, it may be appropriate to audit even more ballots (the mathematics for this are fairly complicated and beyond the scope of this Appendix.) Here we are concerned only with the cost of finding a single error.

The same general mathematical analysis can be applied to an election of any size and margin of victory, simply by substituting the correct numbers. Note that if we spread out the fraud among more precincts (e.g., no more than 20% of the votes in a precinct are bad), then the numbers become somewhat more favorable. However, even then something like 8% of the precincts must be audited.

A Ballot-Based Audit Now let's consider a ballot-based audit. This time we start by picking a single ballot at random and comparing it to the electronic results. As before, if it doesn't match, we've found a problem, but it's quite likely that even if the election was tampered with the first ballot we select will be good. In particular, there are 10,000 bad ballots, so our chance of selecting a good ballot is 990000/1000000, which equals 99%. In other words, selecting a single ballot at random has just as high a chance of detecting fraud as selecting a single precinct at random! Counterintuitive but true.

Now, imagine we select another ballot out of the remaining pool of 999,999 ballots. We have a 989999/999999 chance of picking a good ballot. So, the probability of picking two good ballots becomes:

$$\frac{990000}{100000} \times \frac{989999}{999999} = 0.9801 \qquad (3)$$

Note that this value is just a shade higher than the chance of finding two bad precincts. Why? Because after auditing one good precinct, we have removed about 1% of the good precincts but after auditing one good ballot, we have nearly all the good ballots still to audit! Nevertheless, we've done almost as well with only a tiny fraction of the effort.

If you follow this chain of logic to its conclusion, we find that in order to have a 99% chance of detecting this fraud, we only need to audit 459 distinct ballots, by comparison to 368 precincts with 1000 voters each: 368,000 voters. In other words, ballot based auditing is over 80 times as efficient in this case.

The exact numbers vary with a given election, but the general flavor remains the same: ballot-based auditing is almost always vastly more efficient than precinct-based auditing.