Weight, Weight, Don't Tell Me: Using Scales to Select Ballots for Auditing

Cynthia Sturton[1], Eric Rescorla[2], and David Wagner[1]

[1]*University of California, Berkeley*
[2]*RTFM, Inc.*

Abstract

Ballot-based auditing offers a much higher level of statistical confidence for any given number of ballots counted than does precinct-based auditing. Unfortunately, it also comes with the problem of efficiently finding any particular ballot so that it can be audited. Previous work on ballot-based auditing has required modifying the ballots to add a serial number which could be used for indexing. We describe a method for using scales and ballot weight to quickly index into a stack of ballots. Preliminary experiments suggest that this method may be a practical alternative that is compatible with existing hardware.

1 Introduction

Post-election audits play an important role in ensuring the reliability, accuracy, and security of elections conducted with the aid of untrustworthy computers. However, while audits provide a way to assure that votes have been counted correctly, auditing can consume a significant amount of time and resources.

Current practice is to use *precinct-based audits*, in which a random sample of precincts is selected and then the ballots cast in those precincts are manually recounted; or *machine-based audits*, in which a random sample of machines is selected and then the votes cast on those machines are manually recounted. In either case, one samples by batches. Unfortunately, the statistical effectiveness of auditing is, roughly speaking, a function of the number of batches (e.g., precincts) selected, *not* a function of the total number of ballots recounted. At the same time, the workload is roughly proportional to the total number of ballots recounted. For instance, if we want 95% confidence that at least 99% of the batches were counted correctly, we need to randomly sample and recount about 300 batches of ballots—regardless of the number of ballots per batch [1]. Therefore, reducing the batch size improves the efficiency of audits.

A number of other authors Neff [9], Johnson [7] and Calandrino et al. [2, 3] have proposed "ballot-based" auditing systems: those in which the unit of auditing is a single ballot. In general, each ballot is identified by its index into a stack of ballots (e.g., the 7th ballot in the 3rd stack of ballots). Ballots are sampled by choosing a random ballot out of the universe of ballots and finding the appropriate ballot in the appropriate stack, which is then compared against the corresponding electronic record. However, these systems lack a backwards-compatible method for ballot selection. The difficulty lies in finding the paper ballot that corresponds to the particular electronic record (randomly) chosen for auditing; previous systems required special equipment for printing serial numbers on the ballots plus a brief examination of every ballot to check that the serial numbers were printed correctly.

Our proposal eliminates the need for special equipment while preserving the efficiency benefits of ballot-based auditing. Instead of printing serial numbers on the ballots, we observe that if ballots all have approximately the same mass it is possible to use weight to determine the size of a stack of ballots. This lets us index into a large stack of ballots by pulling approximately the right number of ballots from the stack and then quickly homing in on the correct index with binary search-like methods, iteratively adding or removing ballots until the correct ballot is found. In this way, we can efficiently retrieve the ballots in our random sample so they can be verified against the corresponding electronic records. Another way to think about this is as having an implicit sequence number which is read using the scale.

2 Problem Statement

Our focus is on central-count systems, where ballots are scanned at election headquarters (rather than at the polling places), so that ballots can be stored in the same order they are scanned. Put another way, our proposal can be viewed as an auditable way of scanning and counting a collection of ballots at a central location under the control of election officials. Our protocol consists of two phases: (i) scanning the ballots, (ii) auditing the electronic ballot images. The core of our proposal lies in the auditing phase.

We seek an auditing protocol that meets the following requirements:

- *Ease of use.* We want the auditing protocol to be simple and easy for election officials to conduct, and easy for observers to understand and monitor. This will reduce the likelihood of human error occurring during the audit.
- *Efficiency.* We want to minimize the time and resources required to conduct an audit.
- *Software-independence.* Our audit process must not place any reliance upon the correct functioning of any computer or other complex technology [11]. The purpose of our protocol is to verify that the election equipment has counted the votes correctly, so it must be possible for an election official or observer to see for themselves that the vote count is accurate.
- *Support for risk-limiting audits.* The audit process must be able to detect any error or fault in the election equipment, whether accidental or deliberate, if its magnitude is large enough to change the outcome of an election. Our process must be compatible with risk-limiting post-election audit [12, 6]. In other words, it must be possible to upper-bound the probability that an outcome-changing error is not detected by the audit.
- *Transparency.* Our audit process should be accessible and understandable by any observer. After the audit, the observer should have sufficient evidence to confirm the election count for themselves, without trust in the equipment or the election officials.
- *Compatibility with legacy systems.* The audit process should be compatible with existing voting equipment: jurisdictions should not need to replace their current scanning and tallying machines.
- *Privacy.* Our system should not compromise the secrecy of the ballot. For instance, any information revealed for the purposes of auditing must not reveal how voters voted.
- *Coercion resistance.* Our system should not enable vote-buying or voter coercion where it was not already possible. For instance, any information revealed for the purposes of auditing must not enable voters to prove how they have voted.

We make several assumptions. First, we assume the scanners produce an electronic record of each ballot, and retain these records in the order that ballots were scanned. Many deployed optical scanners (Hart Ballot Now, Sequoia Optech) already retain this information; for those that do not, changes to the scanning software—but not the scanning hardware—may be required. Second, we assume that election officials are careful to store ballots in the order they were scanned. Because we rely on the correspondence between the order of electronic ballot images and the order of the paper ballots themselves, our scheme requires officials to avoid disturbing the order of the ballots, once they are output and stacked by the scanner. We assume that this can be assured through appropriate procedures and processes.

As mentioned earlier, we assume all ballots are centrally scanned. Our scheme could also be deployed in jurisdictions that use precinct-based optical scan (where ballots are scanned at the polling place), by re-scanning all of the ballots centrally, though this would of course require scanning each of those ballots a second time, as described by Calandrino et al.[2].

We focus on ensuring that the collection of paper ballots that is present at the time of the audit matches the electronic records produced by scanners. It is a separate problem to ensure that this collection of paper ballots exactly matches the ballots legitimately cast by voters. Chain of custody issues, while important, are out of scope for this paper; we make no attempt to detect tampering with ballots before they are scanned.

Comparison to prior work. Johnson [8], Neff [9] and Calandrino et al. [2] have previously proposed a variety of methods for ballot-based auditing. All these methods are fairly similar. For concreteness, we describe the method due to Calandrino et al. In their method, the ballots are recounted using optical scan machines augmented with a device that stamps each ballot with a unique serial number as the ballot is scanned. Serial numbers are assigned sequentially,

incrementing by one for each ballot scanned. Alternatively, a separate machine can be used to stamp each ballot with a serial number. The stamped ballot is then scanned and both the serial number and ballot contents are read by the scanner. In either case, the scanner remembers the association between each ballot's serial number and the votes on that ballot. After all ballots have been counted, the voting system produces a list L containing this association, in electronic form; this list is retained by election officials for auditing purposes. Then, auditing involves three steps. First, election officials check that the serial numbers have been printed correctly by quickly looking at the serial number on each ballot. Second, election officials choose an appropriate set of serial numbers at random, retrieve the corresponding paper ballots, and manually inspect each such ballot to check that its contents match what is found in L. This provides assurance (to a given confidence level depending on the number of audited ballots) that no more than a given fraction of the ballots in L have been miscounted. Finally, election officials use independent software to tabulate votes from L.

Calandrino's procedure meets many, but not all, of our requirements. It is efficient, easy to use, and protects voter privacy. It is software-independent and compatible with risk-limiting audits. However, it is not compatible with legacy optical scan machines: in their scheme, scanners must be augmented with a special stamping device. This would require modifications to the hardware of currently deployed voting systems, or design and certification of new equipment for auditing.

Also, their scheme does not aim to provide transparency for observers. Their scheme is focused on enabling election officials to verify that the machines have worked correctly, but observers do not have an opportunity to do the same. It would be possible to extend their scheme to provide transparency for observers, by publishing the list L and adopting transparent random selection procedures—but this comes at a cost. In particular, publishing the list L of ballot images enables vote-buying and coercion: a voter can "mark" their ballot by entering a write-in vote for a unique candidate name in one contest, or by a special pattern of votes in down-ballot contests. Consequently, there is a tension between transparency and coercion-resistance in their scheme.

Our proposal improves upon Calandrino's scheme primarily by eliminating the requirement for special ballot-stamping hardware. Instead, our scheme can be used with deployed optical scan equipment, without any hardware changes or modifications to existing scanners. However, our scheme retains the same tension between transparency and coercion-resistance; we know of no non-cryptographic method for resolving this tension.

3 Our Proposed Procedure

3.1 Overview

The main idea of our proposal is to improve upon Calandrino's scheme, as follows. Rather than printing an explicit serial number on each ballot as it is scanned, we treat the index of that ballot within its stack as an implicit serial number. This index is not printed anywhere on the ballot, but rather comes from its position of the ballot in the stack.

Both procedures provide a way to count a collection of paper ballots with the aid of (untrusted) optical scanners in such a way that the final tally can be verified to represent a correct count of the paper ballots. We can view the audit process as cross-checking three sets of data: the paper ballots, the electronic record of those ballots, and the published vote tally. We start by reviewing the general workflow, which is common to our proposal and Calandrino's scheme, and shown in Figure 1.

We assume that the collection of paper ballots is provided to us, and we do not concern ourselves with where they came from. We also assume the paper ballots have been shuffled in advance to prevent linking ballots to vote order—this is also required in Calandrino's scheme and in any scheme where we publish the CVRs in scanned order. Our scheme consists of 7 steps.

1. *Scanning.* The collection of ballots is divided into conveniently sized batches. The batches do not need to be all of the same size, but in practice we anticipate that each batch will contain 100–250 ballots (this limitation is discussed in Section 5.1.1). For instance, a batch might hold all of (or a subset of) the ballots cast within a single polling place, or a subset of absentee ballots. Election workers feed one batch at a time into the optical scan machine. We assume that the optical scan machine outputs the paper ballots in a stack. For each batch, election officials collect the stack of scanned ballots and place them in a box for storage, taking care not to disturb the order of the ballots, and label the box with the batch number.

   Scanning a batch of n ballots produces a sequence of electronic ballot images $b_1, b_2, \ldots, b_n$, where each ballot image records all of the choices (votes) found on a single ballot. We

assume that these electronic ballot images are produced in the order that ballots appear in the output stack, so that the ith electronic ballot image b_i corresponds to the ith paper ballot in that stack.

2. *Tallying.* Election management software collects up all of the votes and produces a list of vote totals, showing the total number of votes cast for each candidate. The vote totals are published. The software also prepares a list of electronic ballot images $b_1, \ldots, b_n$ for each stack of ballots. These images can either be real images, as in the Humboldt Election Transparency Project[1], or cast vote records which merely show the choices selected for each contest. The latter is far more convenient for our purposes, because what we are trying to audit is ultimately the scanner's interpretation of the ballot. This information is recorded onto write-once media (e.g., CD-ROM) in an open electronic format (e.g., comma-separated values), and copies are distributed to auditors[2]. At this point, the software is committed to a particular vote count, and the audit can begin.

3. *Tally verification.* Auditors verify that the published vote total for each candidate exactly matches the number of votes for that candidate found among the electronic ballot images provided to auditors. This step should be done using independent software chosen by the auditors, or performed using multiple independent software implementations. If any discrepancy is found, the system fails the audit and auditors declare that they were not able to verify the correctness of the published vote totals.

4. *Stack size verification.* For each stack of ballots, election workers retrieve that stack of paper ballots, count the number of ballots in the stack, and verify that this matches the number of electronic ballot images for that stack. If any discrepancy is found, the system fails the audit. Note that this step involves counting only the number of paper ballots in a stack, not counting the number of votes for any candidate. We elaborate below (see Section 3.2) on how this can be done efficiently.

5. *Random selection.* We select a random sample of the electronic ballot images, using a procedure for verifiably random selection that can be seen by observers and auditors to be fair and unbiased [5, 4]. If there are m stacks, with n_i ballot images in the ith stack and a total of $N = n_1 + \cdots + n_m$ ballot images, then this process involves a random selection from the uniform distribution over all N ballot images. The indices of the selected ballots are published.

6. *Ballot retrieval.* Election workers retrieve the paper ballots corresponding to the indices selected in the prior phase. This is done in two steps: for each selected ballot, we retrieve the box containing the stack it is contained in; then we index into that stack to retrieve the particular ballot we are interested in. Assuming that boxes are clearly labelled, finding the appropriate stack should be relatively straightforward. In section 3.3 we explain how to index into a stack of ballots efficiently.

7. *Manual comparison.* Election workers manually inspect each selected paper ballot to compare the marks on it to the corresponding electronic ballot image. It is important that every contest on the ballot be inspected and compared; this provides a reasonable way to detect human error in the indexing process.

   If few mismatches are found during this step, we have strong evidence that the electronic ballot images match the paper ballots, to within an error bound that can be calculated with standard statistical methods [1, 10, 7].

3.2 Counting stack sizes

To verify stack sizes efficiently, we need some way to quickly count the number of paper ballots in a stack. We take advantage of the fact that ballot weights are approximately constant from ballot to ballot: we use a scale to measure the weight of the entire stack of ballots, and then divide by the weight of a single ballot to obtain an estimate of the number of ballots in the stack. This can be done fairly

[1] `http://humtp.com/`

[2] Our scheme is agnostic about who serves as an auditor. Election officials or third parties may play the role of auditors, if they wish to check that the scanners worked correctly. Alternatively, if we wish to allow members of the public to verify for themselves that the scanners worked properly, ballot images can be published on the Internet or distributed to all observers in attendance during the audit process. Note that auditors receive a list of ballot images, so a voter who marked their ballot in advance may be able to prove how they voted to an auditor. Therefore, auditors must be trusted not to coerce voters or buy votes. Officials might choose to make ballot images available to the general public in the interests of transparency, and accept the risk of vote-buying and coercion. As mentioned earlier, there is a tension between transparency and coercion-resistance in our scheme.

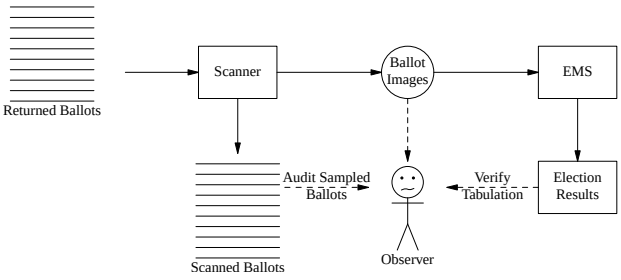

Figure 1: The audit process: the collection of voted ballots are scanned, the election management system produces a tally from the electronic record and the electronic record is made public. The observer audits samples from the scanned ballots against the electronic record to verify the scanner produced the correct electronic record. The observer produces a tally from the electronic record to verify the EMS-produced correct tally.

quickly—more quickly than manually counting the number of ballots one by one.

The mathematics are simple. Let μ denote the average weight of a single ballot. Given a stack of weight W, we can estimate the number of ballots in the stack ($\hat{n}$) by dividing and rounding to the nearest integer, i.e., $\hat{n} = \lfloor W/\mu + \frac{1}{2} \rfloor$. Note that in practice μ will not be known precisely; instead, we will need a calibration step to compute an estimate $\hat{\mu}$ of μ. For instance, officials might gather 100 ballots, weigh those 100 ballots, and divide by 100 to get $\hat{\mu}$. Fortunately, the calibration step only needs to be done once per audit.

This process can be greatly simplified with the help of a counting scale, which produces an item count as well as total mass. Counting scales are readily available commodity items: we obtained the counting scale used in our experiments for around $230. With a counting scale, calibration is simple: we place a large stack of a known number of ballots on the scale, wait for it to stabilize, and then enter the number of ballots into a keypad on the scale. The scale then derives an estimate $\hat{\mu}$ of the weight of a single ballot. The larger the stack used for calibration, the more accurate $\hat{\mu}$ will be. In our experiments, we used a stack of 350 ballots to calibrate the scale. We recommend that calibration be performed after the election, shortly before the audit, under the same environmental conditions as will occur during the audit itself[3].

Once the counting scale is calibrated, counting the number of ballots in a stack is easy: we place the stack on the scale, wait for it to stabilize, and the scale shows its estimate of the number of items on a digital display. In summary, we have a simple and efficient way to check that the number of paper ballots in each stack matches the number of electronic ballot images for that stack (step 4).

3.3 Retrieving a specific ballot

A counting scale also gives a convenient and efficient way to index into a stack of ballots (see step 6). In particular, given an index i and a stack of paper ballots, one can use a counting scale to quickly find the ith ballot in the stack.

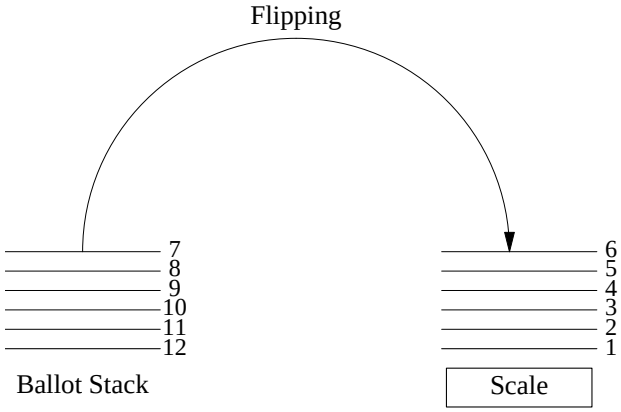

Figure 2: Flipping ballots onto the scale

We found that the easiest way to do this is to place the full stack of ballots on the table next to the scale, with the ballots ordered top to bottom, and "flip" batches of the ballots onto the scale until the scale reads the appropriate number. For example, if we are looking for the 100th ballot, we might grab a bunch of ballots from the top of the stack, place those on the scale (flipping them as we do so) and look at the read-out to see how many ballots are currently on the scale. If the scale shows 87 ballots, we might grab a few additional ballots from the top of the stack on the table and check the read-out again. If the readout now shows 105 ballots, we would return several ballots from the scale to the stack on the table. We continue in this way, in a binary search-like fashion, until the scale shows 100 ballots. At that point, the top ballot on the scale is the 100th ballot. As shown in Figure 2, at each step along the way, the ballots on the scale are "upside down" and the ballots on the table are in their original orientation, which makes it easy to preserve the order of the ballots.

[3]See also Appendix A for a discussion of several alternatives and justification of this recommendation.

We found in our experiments that this process converged rapidly, with just a few steps. In particular, this process requires many fewer rounds than the $\lg n$ rounds one might expect for a binary search, because we learned to "eyeball" things so that we could pick roughly the desired number of ballots and thus converge towards the desired result. In principle, a similar process could be applied with a regular scale, but the math might slow down the procedure considerably and introduce an opportunity for human errors to be introduced; with a counting scale, the procedure is simple and straightforward.

3.4 Random selection

To assist in random selection of ballots, we recommend that the election management software generate a table to help select a ballot at random. For example, suppose we have three stacks, containing 300, 200, and 160 ballots, respectively. Then we could use the following table for use in random selection:

number	location
1–300	stack 1
301–500	stack 2
501–660	stack 3

Random selection would then consist of selecting a number uniformly at random between 1 and 660, identifying the appropriate stack, and then subtracting to obtain the index within the stack. For instance, the random number 360 corresponds to the 60th ballot in stack 2. This table could be prepared before the random selection process, and auditors could easily verify that it was prepared correctly.

If we are using dice to generate random numbers, this can be tweaked slightly to ease the process. With the same example as above, we might generate a slightly different table:

number	location
000–299	stack 1
300–499	stack 2
500–659	stack 3
660–999	re-roll

Then we could roll three ten-sided dice to choose a random number between 0 and 999, look up the entry in the table, and re-roll if necessary (if the chosen number was in the range 660 to 999).

Once the ballots to audit have been selected, it is probably worth sorting them by stack and index so that we don't need to repeatedly find the same stack. In addition, if we need to select two ballots out of the same stack and we do so in order, then we can checkpoint the stack at the first ballot and then just add the required number of ballots, rather than having to look through the entire stack each time. Sorting by stacks also enables parallel auditing if we have multiple scales.[4]

3.5 Practical considerations

It is crucial for this scheme that, once scanned, the ballots stay in the same order. If a stack gets shuffled or re-ordered after being scanned, there is clearly no way to rely on a ballot's index in the stack to identify the ballot.

Maintaining ballot order would probably be a good idea in Calandrino's scheme as well: even with the serial numbers stamped on the ballots, if the ballots get shuffled finding the correct ballot becomes a linear search process, which isn't dramatically faster than counting into the stack. However, Calandrino's scheme is much more robust to occasional human error that causes some ballots to be re-ordered, as the ballots can be sorted back into the correct order (although this is of course expensive). In contrast, our scheme has the drawback that there is no direct way to know whether the stack has been shuffled, so our scheme would require strict protocols to ensure the stacks stay in order; if the stacks get out of order, then almost every ballot we check will not match and the audit will fail. Our method of "flipping" stacks onto and off of the scale from a large stack should help maintain the order during the indexing operation, but still, considerable care will be required whenever ballots are handled.

Our scheme also requires that batch sizes be small enough to to avoid selection errors. In our experiments we used a stack of 350 ballots without error. However, our measurements (see Section 5.1.1) suggest that in some cases stacks greater than 200–250 may show excessive errors and so it is safer to stick with stacks around this size until we have more experience with actual error rates. We believe this limitation is not serious, as larger stacks are already fairly unwieldy. Note that these limits are not exact: the easiest procedure is to estimate the thickness of a stack of 200–250 ballots and then divide stacks by eye.

3.6 Error handling

Because this process involves human processing, mistakes may occur. It is important to have a process that can tolerate occasional human error. Accordingly, we suggest error handling procedures for

[4]We owe this discussion to an anonymous reviewer.

two particular kinds of issues that we expect may arise:

- *Discrepancies in stack sizes.* Suppose that during step 4, we weigh a stack and find that the number of paper ballots in the stack appears to be different from the number of electronic ballot images for that stack. What should election workers do? One possibility is to divide the stack of paper ballots into a few smaller chunks, weigh each chunk, and then add up the number of ballots in each chunk. Because the weighing process is more accurate for smaller stacks, this provides a quick check that may eliminate some kinds of error (see Section 4). If the discrepancy persists, the next step is to manually count the number of ballots in the stack, using any of a number of standard procedures—e.g., split up the stack into groups of ten and count the number of groups by hand. If the discrepancy still persists, it is likely that something went wrong with the scanning or handling of paper ballots; election officials will need to investigate, determine what went wrong, and take corrective action. After the paper ballots or electronic ballot images are corrected, the audit may need to be re-started from scratch.

- *Mismatch between the paper and electronic record.* Suppose that during step 7, we retrieve a paper ballot and discover that the votes on it do not appear to match the electronic ballot image. What should we do?

  There are two major possibilities: (1) our weighing procedure has failed and we picked the wrong ballot out of the stack and (2) we have the right ballot (though the stack may have been shuffled) and the electronic records truly do not match. (We discuss the situation where we picked the wrong ballot but by coincidence it matches the electronic record in section 5.1.2.) We can eliminate the first possibility by recounting the stack with our chosen ballot at the top, either manually or by dividing it into smaller stacks. Continuing with our previous example where we are looking for the 100^{th} ballot, the recounting protocol would work as follows. Suppose the counting scale tells us that we have 100 ballots on the scale, but when we look at the top ballot it does not match the electronic record for the 100^{th} ballot. We would then invoke the recounting protocol:

  1. Flip the 100 ballots back onto their original stack.
  2. Using the procedure for retrieving a specific ballot, remove the top 50 ballots from the stack and set them aside.
  3. Again, use the procedure for retrieving a specific ballot to create a stack of 50 ballots on the scale. The top ballot should be the 100^{th} ballot, ready for comparison against the electronic record.

  Since the weighing procedure for retrieving a specific ballot is more accurate for smaller stack sizes, dividing the stack into smaller sub-stacks will reduce the chance of error in selecting the correct ballot. Of course, depending on how large the ballot index that you are looking for is, you may wish to divide the stack into more than two substacks. Furthermore, it is not necessary to subdivide the stack evenly. In our example, the second step could be done by eyeballing roughly 50 ballots. If the first substack that is placed on the scale and then set aside has 54 ballots, then in the third step the 46^{th} ballot would need to be retrieved.

  If the ballot index is small enough, it might be easier to forego using the scale in the recounting protocol and instead perform a manual recount. In this case, the 100 ballots would be flipped back onto their original stack and then the auditor would simply count off the first 100 ballots. However, some degree of transparency is lost as observers are unable to easily see how the ballots are being counted. Therefore we suggest starting with divided weighings.

  For handling an actual mismatch between a ballot and electronic record there are several steps we can take to deal with the possibility of an occasional scanning error. It is known that optical scan machines may fail to interpret some kinds of marks in the same way as a human would—particularly in the case of ambiguous or improper marks or miscalibration of the scanner. Consequently, we can expect a small probability that any given ballot may fail to match due to this kind of occasional scanning error. We anticipate that this kind of scanning error will likely be readily identifiable, because it may manifest as a mismatch in only a single contest or a visibly ambiguous mark. If this kind of expected scanner error is suspected, we recommend recording this as a mismatch and continuing on with the audit.

  If the indexing appears to be correct, and there remains a mismatch that does not seem to be at-

tributable to ordinary scanner error, there are several possibilities. For ease of implementation, we suggest recording a mismatch and continuing on to the next selected ballot. Other possibilities might be to examine adjacent ballot images to see if they match the paper ballot, in case of an off-by-one error, or to manually count all contests on the entire stack of paper ballots and compare it to a tally of all of the corresponding electronic ballot images. However, the former introduces subtle security issues, and the latter is likely to be time-consuming. It is also probably prudent to separately investigate such errors in order to determine the cause — unexplained errors are troubling even if they do not affect the winner of the contest.

The important thing is to record the number of matches and mismatches, so that at the end of the audit we can use statistical methods to evaluate the degree of confidence provided by the audit. Johnson [7] describes methods for escalating ballot-based audits to obtain the appropriate degree of confidence when individual errors are detected. In some cases it may be easier to record a mismatch and move on than to try to laboriously diagnose the cause of a mismatch. However, it is important to note that because our sample size is small one or two mismatches can imply a significant error in the reported totals, so some care should be taken to avoid them.

In general, whenever we detect some kind of error, we can always fall back to manual counting of the ballots. Thus, except for the edge case where we have undetected errors (see Section 5.1.2), one can think of weighing as providing a fast, but potentially imperfect method for finding ballots with hand counting serving as a backup.[5]

4 Sources of Selection Errors

The primary new source of error that our system introduces is that we may select the wrong ballot (typically the one before or the one after). We know of three major mis-selection mechanisms:

- *Scale error.* The simplest kind of error is simply that our scale might report an incorrect value. If the error were large enough, it might cause us to misestimate the size of a given ballot stack.
- *Ballot stack variance.* Ballots are a manufactured product and therefore their weights are not identical, but rather follow some distribution. This means that any given stack might be composed of ballots which on average are heavier or lighter than μ, leading to inaccurate selections.
- *Mis-estimating* μ. As discussed in Appendix A, we generally cannot measure μ directly. Instead, we must estimate μ by weighing some convenient set of ballots of known size and dividing to find $\hat{\mu}$, which is an estimate of μ. This calibration procedure can cause selection errors if different ballot lots have different distributions.

The remainder of this section attempts to quantify the magnitude and impact of these errors.

The first source of error is primarily a cost/convenience issue. Nearly arbitrarily accurate scales and balances are readily available at an equally arbitrarily high price; although we expect far cheaper scales will be adequate. So, while we cannot discount this issue, the latter two issues are of more immediate concern because they cannot be solved by the purchase of better equipment: they present an issue even if we are weighing on perfect scales.

4.1 Ballot Stack Variance

To analyze the impact of ballot stack variance, we start with a simple model in which the weights of ballots are normally distributed with mean μ and standard deviation σ. If we randomly select a sample of n ballots from this distribution, we expect the sum of the ballot weights $W = \sum_{i=1}^{n} w_i$ to also be normally distributed with mean $n\mu$ and standard deviation $\sqrt{n}\sigma$. Let $\hat{n} = \delta(W)$ be the estimated number of ballots given weight W. Now $\delta(W)$ is only correct if $\mu(n - \frac{1}{2}) \leq W < \mu(n + \frac{1}{2})$. The width of this band is constant, whereas the variation in W scales with $\sqrt{n}$. Thus, as the number of ballots increases, variation in ballot weight accumulates and more of the probability mass of the distribution falls outside the correct band $n\mu \pm \mu/2$.

Figures 3 and 4 show this visually: the curve shows the probability density function for total stack weight for $(\mu = 1, \sigma = .015)$. The shaded region represents the fraction of total stack weights for which $\delta(W)$ is correct. Because the density function is broader for $n = 500$ than for $n = 100$, far more of the probability mass falls outside the correct region. The exact error rate is of course dependent on the ballot weight distribution, in this case the σ/μ ratio.

[5] We are grateful to an anonymous reviewer for this point.

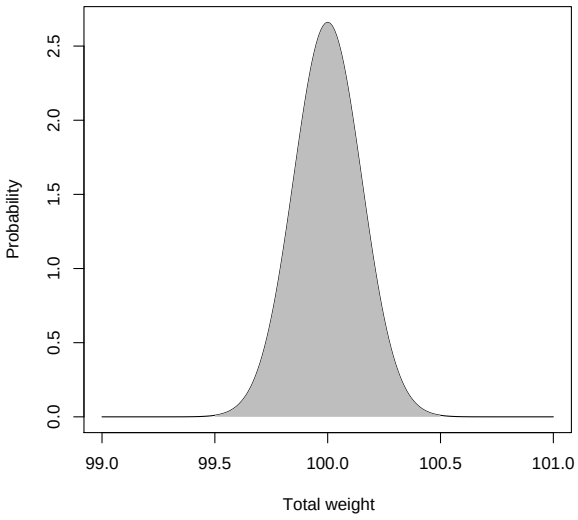

Figure 3: Distribution of total weight: $n = 100$

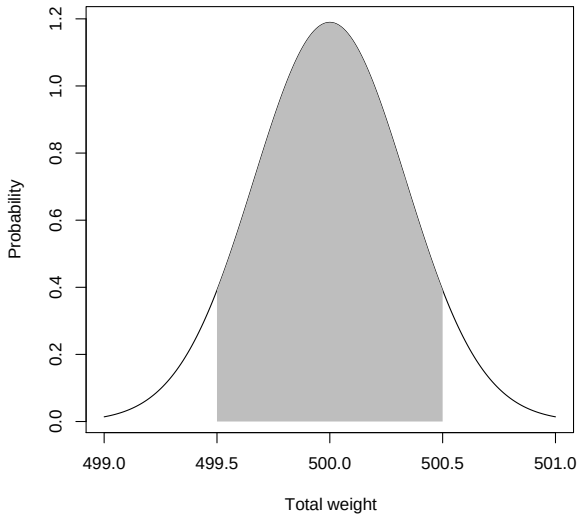

Figure 4: Distribution of total weight: $n = 500$

4.2 Mis-estimating μ

A related problem is errors in determining $\hat{\mu}$. Because we compute $\hat{\mu}$ from one set of ballots and then use that to estimate the size of another, possibly disjoint, set of ballots, the value of $\hat{\mu}$ determined from our calibration set may not be exactly the same as the mean weight of the ballots we are actually trying to count. Because this is a systematic rather than a random error, even very small differences between μ and $\hat{\mu}$ can have a significant impact on the error rate. This is discussed further in Section 5.1.1.

In order to control for this effect, we recommend that auditors sample multiple batches to determine $\hat{\mu}$. In addition, the procedure of verifying that the total batch weight is consistent with the reported number of ballots in the batch provides a fairly sensitive test for weight discrepancies.

5 Empirical Results

In order to evaluate the magnitude of these effects for real ballots, we obtained two sets of unvoted ballots from Doug Jones at the University of Iowa. These are leftover sample Optech 4C scan ballots from Maricopa County, Arizona, approximately 18×9.75" in size. We labelled the two sets "A" and "B". We then performed experiments with two scales: one experiment with a counting scale suitable for weighing large stacks of ballots, and another experiment using a high accuracy scale with a small range suitable for weighing single ballots.

5.1 Distribution of Ballot Weights

As discussed above, our principal concern was to determine whether the variation in ballot weights was sufficiently large to produce significant errors. Thus, our first set of measurements was intended to determine the distribution of ballot weights.[6] Accordingly, we individually weighed the first 100 ballots from each box. The weighings were done with a Acculab GS-200 lab scale (range 0-200g, readability .1g). This scale has a rather small platform (5.125" square), compared to the ballot size (18" × 9.75"), with the result that the ballots hang substantially over the side of the platform. We were forced to raise the scale up in order to avoid letting the ballots touch the table and affecting the measurement. In addition, for some ballots the scale was unable to converge on a single value for the final digit (e.g., oscillating between 21.9g and 22.0g). We attribute

[6] Note to scientifically minded readers: in SI units "grams" refers to "mass", which is an intrinsic property of the ballots rather than "weight", which depends on the particular gravitational environment in which the measurements were taken. However, the electronic scales in common use actually measure weight, or rather force, and simply translate it to mass based on standardized assumptions about the gravitational field. In accordance with lay practice, we will use the term "weight" and report it in grams rather than Newtons, the SI unit of force.

this to a combination of a ballot weighing roughly in the middle and air current-induced instability. In such cases we arbitrarily assigned the weight to halfway between the two values. Finally, in some cases we took several measurements (e.g., before or after changing the battery on the scale), in which case we averaged them.

Figures 5 and 6 show the distributions within each batch, with the bins being .1g wide. In other words, the values 21.10 and 21.15 are in the same bin. While the distributions appear fairly different, their mean weights differ by less than .1%, which is statistically insignificant (t-test $p = .35$). However, the variance is significantly different (Levene test, $p < .01$). This suggests that there is some systematic difference between the boxes. It is unclear whether the ballots are actually from different distributions or whether our technique varied, as we weighed box A first and may have been more practiced and/or attuned to scale instabilities during the second set of weighings; we saw many more such instabilities in box B (29) than box A (8). Note that even box B, which appears to the eye to be less normal, has a standard deviation of less than 1% of the mean ballot weight.

As a double check, we weighed the first 350 ballots in each stack using our lower resolution, higher range scale, and the means are rather closer, suggesting that we may be seeing the cumulative effect of small weighing errors in each ballot. In future, we would like to take individual measurements of more ballots with a higher resolution scale in a more controlled environment (see Section 7).

5.1.1 Projected Selection Error

Given the above distributions of ballot weight, it is possible to estimate the error rate in ballot selection, assuming that the scale we are using to count out the ballots is error-free. We model the situation as follows:

- Ballot weights are distributed according to some probability density function $\varphi(w)$ with true mean μ.
- We randomly generate ballots from $\varphi(w)$ and compute the mean, which gives us an estimated μ, $\hat{\mu}$.
- We randomly generate ballots B from $\varphi(w)$. Denote the weight of each ballot in the stack as $w_1, w_2, \ldots, w_n$.

Because the distribution for box B appears non-normal, we decided not to fit a standard distribution but rather to use numerical methods, assuming that the measured distribution is the correct one, at least for that stack of ballots.[7] This reduces to sampling our existing ballot stacks with replacement. We wrote a simulator in Python to model this situation. Our simulator first randomly generates 1000 ballots to compute $\hat{\mu}$ and then generates a stack of 500 ballots. For each position i in the stack, we determine whether we could correctly estimate the stack size. In other words, we check whether $i \stackrel{?}{=} \delta(\sum_{j=1}^{i} w_j)$, assuming that $\mu = \hat{\mu}$. Figure 7 shows the results of this procedure, with each simulation run 100,000 times.

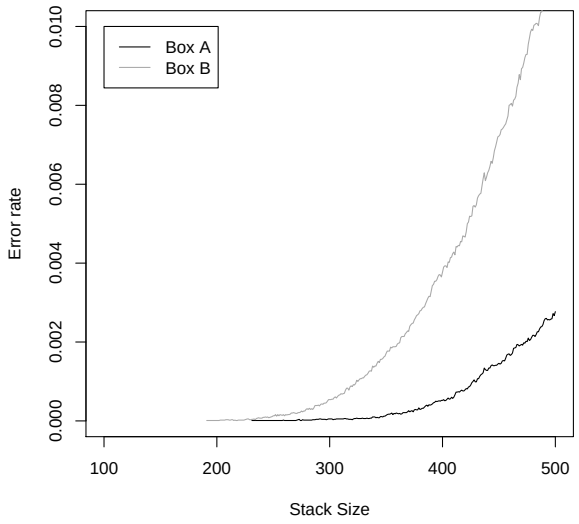

Figure 7: Simulated error rate

We can make several qualitativate observations about this figure. First, the error rate increases rapidly with increasing stack size, which is as we expected from the discussion in Section 4. Second, for stacks of below about 230 ballots (for box A) and 190 ballots (for box B), our simulations produced no errors at all. This implies that if we're willing to divide our ballots into suitably small stacks, we can reduce the error rate arbitrarily. Third, the error rate for box B is much worse than the rate for box A: even with stacks of 500 ballots, box A's error rate is $< .5\%$, while a similar stack with box B has an error rate exceeding 1%.

To evaluate the impact of mis-estimating $\hat{\mu}$, we also ran simulations where we used one box to compute $\hat{\mu}$ and then simulated selecting ballots from the other box. Figure 8 shows the results of four such simulations. We computed the mean in two

[7] Simulations using normal distributions with the same μ and σ as these distributions produce similar results.

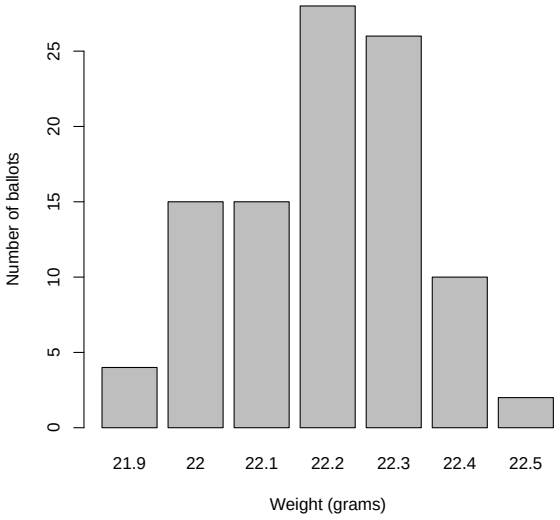

Figure 5: Ballot weights: box A

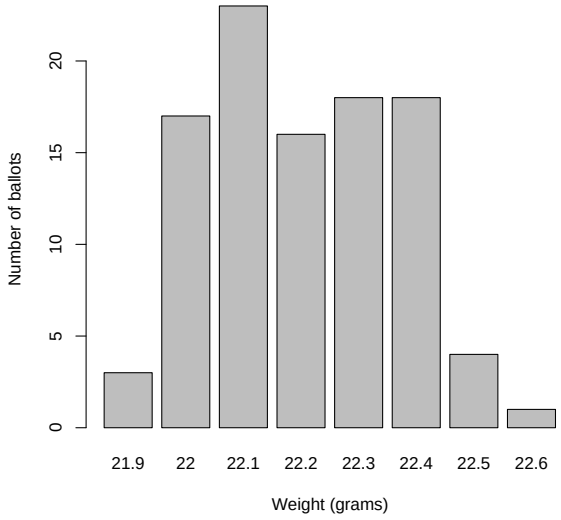

Figure 6: Ballot weights: box B

ways, both directly and by the sampling procedure described in the previous section. from the theoretical analysis in Section 4, errors in $\hat{\mu}$ result in significantly worse error rates in these cross-box simulations than we saw in the previous section. Moreover, using the true mean (even from the wrong box) produces somewhat better results than the resampled mean, because we do not have to contend with the variation due to sampling. Arguably, using the true mean is more relevant here: we can think of both boxes as being drawn from the same large pool of ballots with Box A being used as our calibration sample and Box B as being our auditing sample; simulating resampling again introduces more error than we would expect in the real world. Note that this difference is more pronounced if the means are closer (as our overall weighings suggest they might be) and so systematic effects are smaller.

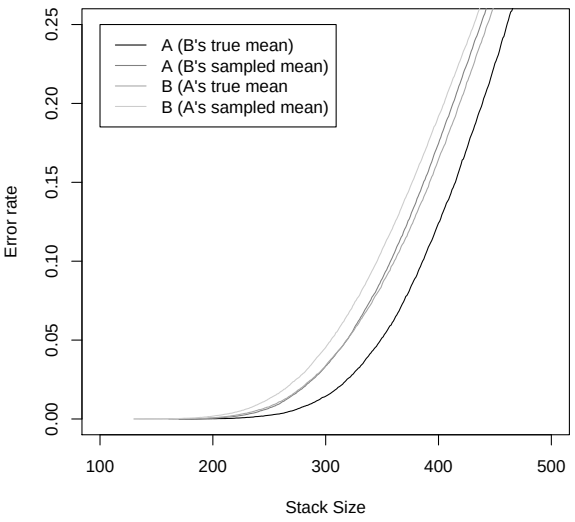

Figure 8: Simulated error rate: cross-box sampling

5.1.2 Impact of Mis-Selection

The above data suggests that while selection errors will be reasonably rare, they will not be nonexistent. If we are trying to select ballot i and actually select ballot $j \neq i$, we need to consider four major cases:

- i scanned correctly but the contents of j do not match i.
- i scanned correctly and by coincidence the contents of j do match i.
- i scanned incorrectly and the contents of j do not match the scan results for i.
- i scanned incorrectly and by coincidence the contents of j do match the scan results for i.

Only the first and fourth cases matter; in the second and third cases the results are the same as if we

had actually selected i. However, in the first case we get a "false reject": a report of an error where there was none. In the fourth case, we get a "false accept": a report of success where in fact the ballot was scanned incorrectly.

As described in Section 3.6, dealing with false rejects does not seem particularly problematic, as long as the rate is not too high: if the manual evaluation of a ballot does not match the electronic records, it is natural to go back and verify that the correct ballot was selected by manual counting. This is inconvenient but not particularly expensive as long as it does not occur too often and the stacks are sufficiently small.

False accepts are more problematic, because they are harder to detect. If the false accept rate is reasonably small, we can compensate by sampling fractionally more ballots, as described by Johnson [7], in the case where the error rate e is relatively small, we can simply compensate by multiplying the number of ballots to audit by $1/(1-e)$. We can compute a conservative estimate for the false accept rate by treating the misselection rate for maximal sized stacks as if it applies to stacks of all sizes and by assuming that all misselections are false accepts. However in practice, misselections are much more likely to result in false rejects than false accepts. Data from the 2006 Sarasota County General Election indicates that the chance of two randomly chosen ballots having identical contents is less than 0.0005: over 99.9% of random selection errors will result in false rejects rather than false accepts. (Note that this calculation depends on auditing the entire ballot. If single races are audited or the ballot is unusually short, then the chance of duplication is naturally higher. It would be interesting to look at other sets of ballot data to find projected matching rates.)

Of course, an attacker might attempt to maximize his chances of success by duplicating electronic record $i+1$ (or $i-1$) onto record i, with the hope that the misselection will work in his favor, but even then it is equiprobable that the misselection will cause the wrong ballot in the opposite direction to be selected. In general, stack sizes must be chosen so that the overall error rate (and hence the false reject rate) is no higher than a few percent. Otherwise, explicit error handling becomes too expensive. Thus, even if we were to assume a worst case scenario that in some election all errors are false accepts, we would only need to oversample by a few percent in order to compensate.

5.2 Selection Experiments

We also tried an end-to-end test using our counting scale (Virtual Measurements & Control VW-330A-C 50kg x 0.002kg) and 350 ballots from box A. We calibrated the scale with the same stack and then one of us (EKR) generated random indices while another (CKS) used the scale and our ballot retrieval method to find the correct ballot. Over the course of 20 trials, the longest time it took to find the ballot was 31 seconds (on an early trial), with all remaining trials taking less than 30 seconds. In each case the ballot retrieval method resulted in selecting the correct ballot.

5.3 Tampering with Ballot Weights

Our procedure relies on the approximate constancy of ballot weights. If someone could significantly affect the weight of the ballots, she would be able to disrupt the auditing scheme. One way to change the weight of a ballot is to tear off some part of it. This might be done deliberately or accidentally, especially in the situation where ballots have a perforated stub that should be removed by the voter. Leaving this stub in place or tearing off a portion of the ballot while removing the stub would affect the total weight of the ballot. Some jurisdictions already take care to remove the stubs out of concern for jamming the scanners, but our method would require that all stubs be removed prior to auditing. Ballots with the stub still attached are easily detected by the natural method of aligning the ballots at the bottom and looking for over-long ballots. It is harder to detect ballots which were torn in stub removal. In our experience as pollworkers, this is rare, but if future experience showed it to be a problem it might be possible to modify scanners to have them reject damaged ballots.

Another way to alter the weight of the ballot is by adding mass, for example by getting the ballot wet. To test this out, we wiped down one ballot with a damp sponge and were able to increase the weight of the ballot by 4.6g, roughly 20% of the ballot weight. Three such ballots would be enough to introduce an off-by-one error when we weigh the stack (due to rounding). We then soaked the ballot in water; this increased the weight by a total of 5.9g. This is a significant amount; on the other hand, a wet ballot might be easily noticed by election officials or might have a chance to dry out before the auditing procedures begin. If, however, a ballot can be wiped down with a solution such that additional mass remains after the water has evaporated, that could affect our indexing scheme and go unnoticed. If a mis-

selection occurs, a false reject or false accept results; as discussed in section 5.1.2, a false reject is much more likely than a false accept. In addition, a single ballot can only affect the indexing of a single stack of ballots; to introduce widespread error, many ballots across many precincts would need to be altered. Therefore, in the worst case, if a large number of ballots, distributed across many precincts, become altered, our scheme reduces to manual counting to find the sampled ballots.

6 Discussion

In Section 2, we laid out eight stated goals: ease of use, efficiency, software-independence, support for risk-limiting audits, transparency, compatibility with legacy systems, and privacy. Our system meets some, but not all of these, as discussed below.

- *Ease of use.* The pseudo-binary search method is straightforward and easy to use. In our experiments we were able to find the correct ballot in a stack of 350 rapidly. Our system of stacking the ballots next to the scale and "flipping" more ballots onto or off of the scale as needed ensures that the stack remains in order.

- *Efficiency.* The equipment needed for our scheme is modest: a simple counting scale. The one we bought cost $230. The human resources are also modest. For each ballot in the random sample, we must: (i) find the appropriate stack of ballots, (ii) select the paper ballot at the appropriate index in that stack, and (iii) manually audit the selected ballot. If the stacks are labelled and organized appropriately, we anticipate that it may take a minute or so (this depends on the number of stacks, obviously) to locate the appropriate stack, which corresponds to several hours if 300 ballots are audited. Finally, Stevens estimates that it takes 6 seconds per contest to manually count a single ballot [13]; in an election with 30 contests, this corresponds to 3 minutes per ballot, or 15 hours to recount 300 ballots, much more than the cost of finding the ballots using our method.

  Achieving a similar level of statistical confidence from precinct-based or machine-based audits would require recounting a very large number of ballots and a correspondingly large commitment of time and resources. Because of resource constraints, jurisdictions that currently use precinct- or machine-based audits make no attempt to achieve 95% confidence (or anything approaching such a degree of effectiveness).

- *Software-independence.* Our scheme does not trust or rely upon correct operation of the scanners or associated equipment. The only machinery introduced by our scheme is a counting scale, which can be checked for accuracy before the audit procedures begin. We assume that the scale is a simple device whose failure modes are well-understood and are consistent from weighing to weighing, so that any failure of the scale will become immediately obvious during this preliminary check. Because scales are generally computer controlled, in principle a malicious insider with access to the scale might tamper with one to cause it to read out incorrectly: the most powerful attack we know of would be to install a radio receiver which would let an external attacker override the scale readout. This threat could potentially be mitigated, at a significant cost in terms of convenience, by using a manual scale. Another potential mitigation would be to use the counting scale for ballot selection but then a manual scale as a double-check on the selected stack position.

- *Support for risk-limiting audits.* Our approach is well-suited to use of risk-limiting audits. Once the scanning is complete and a tally has been committed, the sample size required to meet the goals of the audit can be determined using standard statistical techniques [1, 7]. Our protocol focuses on selecting a specified number of ballots, not on calculating the required sample size or confidence level.

- *Transparency.* Our system is largely, but not completely, transparent. Any observer with a view of the scale can verify that it was calibrated and that the scale is reading the correct weight and count for the desired ballot. One potential threat to transparency is the manipulation of the ballots themselves. Once the election official handling the ballots has determined the correct stack size, he might be able to substitute another ballot (such as the next ballot down) for the actual audit. Both of these attacks only work well if there is a nearby ballot which duplicates the electronic record, but this is a natural type of attack for a malicious scanner to mount.

- *Compatibility with legacy systems.* Our system will work with any optical scanner equipment that scans paper ballots, produces an electronic, numbered list of those ballots and their content, and outputs the scanned ballots into a stack in the same order as the electronic list. We require

no special equipment for re-scanning or stamping the ballots.

- *Privacy.* Assuming that the ballots are shuffled before being scanned, our system protects the secrecy of the ballot.
- *Coercion resistance.* Unfortunately, our system fails to achieve this criterion. If we want to allow independent observers to verify the election outcome, we need to provide them with the list of electronic ballot images before the audit begins. However, this list of ballot images enables voters to prove how they have voted, either with a special write-in name or with pattern voting in down-ballot contests. When voters can prove how they voted, they can be coerced, or their votes can be bought. We do not have a satisfactory resolution to this tension between transparency and coercion-resistance.

While ideally we would meet all eight goals, we have improved the best previous ballot-based auditing systems by proposing a system which is compatible with existing legacy systems. The major remaining deficiency is the tension between privacy and coercion resistance. Designing a simple method of resolving this tension is an open research problem.

7 Future Work

While the data presented in this paper suggests that our proposal may be viable, significant amounts of further work are required in order to determine whether it is really practical. The most important open question is to determine the level of variation of the weights of real voted ballots, perhaps by collaborating with a registrar of voters to measure such a set of ballots. In addition, it would be useful to use a higher resolution scale than the one we have—at the last minute we were able to obtain such a scale but the initial experiments we performed exhibited too much drift (about .1g variation over the course of weighing 100 ballots) to give us better data than the weighings reported here. We have contacted the manufacturer but do not yet have a resolution of this issue.

Another open question is how well our proposal works in the field, including finding the right stack, having untrained operators do weighings, etc. Before the scheme could be deployed it would be important to do an end-to-end study involving selection and perhaps auditing of live ballots. This would help to validate whether our initial measurements of time to find a ballot scale to real-world use.

Finally, it would be important to determine whether ballot order can be preserved between scanning and auditing. One experimental approach might be to select a number of boxes of stored ballots and then re-scanning them (perhaps using technology from the Humboldt ETP) to compare the rescans to the original CVRs.

All of these experiments are likely to require some measure of cooperation from voting officials.

8 Summary

Previous authors have demonstrated that ballot-based auditing can provide equivalent levels of statistical confidence to precinct-based audits while requiring the audit of far fewer ballots. We have described a method for ballot-based auditing that can be applied to legacy ballot-scanning equipment without requiring hardware changes to existing voting systems. While we have presented preliminary evidence to suggest that the method is practical, more research is needed to answer some of the open questions that remain. An empirical end-to-end evaluation of a deployment of our scheme would provide information about the effort required to form the stacks of ballots, the feasibility of keeping scanned ballot stacks in order, the ease with which the correct stack can be located, and a quantification of the efficiency of our system as compared to precinct-based auditing systems. We hope that this work will contribute to the study of how to audit elections cost-effectively.

9 Acknowledgements

We thank Doug Jones for sharing with us sample ballots; this work would not have been possible without his assistance. Thanks to David Dill for providing us with data from the Sarasota election. Thanks to the anonymous EVT reviewers for their helpful comments. This research was supported by NSF CNS-0524745 and by a University of California Chancellor's Fellowship.

References

[1] Aslam, J. A., Popa, R. A., and Rivest, R. L. On estimating the size and confidence of a statistical audit. In *Proc. 2007 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT '07)* (2007).

[2] Calandrino, J. A., Halderman, J. A., and Felten, E. W. Machine-assisted election auditing. *USENIX/ACCURATE Electronic Voting Technology Workshop 2007* (Aug. 2007).

[3] Calandrino, J. A., Halderman, J. A., and Felten, E. W. SYSTEM AND METHOD FOR MACHINE-ASSISTED ELECTION AUDITING. US Patent Application 20090037260, August 2007.

[4] 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).

[5] 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).

[6] ElectionAudits.org. Best Practices: Risk-Limiting Audits. `http://www.electionaudits.org/bp-risklimiting`.

[7] Johnson, K. C. Election certification by statistical audit of voter-verified paper ballots, Oct. 2004.

[8] 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`.

[9] Neff, C. A. Election confidence—a comparison of methodologies and their relative effectiveness at achieving it (revision 6), Dec. 2003.

[10] Rivest, R. L. A simple rule of thumb for election audit size determination. Tech. rep., Massachusetts Institute of Technology, Nov. 2007.

[11] 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`.

[12] Stark, P. B. Risk-limiting post-election audits: P-values from common probability inequalities. Tech. rep., University of California at Berkeley Department of Statistics, Feb. 2009.

[13] Stevens, A. Hand counting paper ballots. Democracy Fest Annual National Convention, June 2007.

A Alternatives for Determining $\hat{\mu}$

The procedure described in this paper assumes that $\hat{\mu}$ is determined just prior to the audit procedure. This is the most transparent method since any observer can verify that the correct procedure has been observed, however it is not necessarily the most convenient. Two alternatives would be to have the printer provide μ or to determine it prior to the election.[8]

In principle, when the the printer produced a batch of ballots they could also provide their true μ value by weighing the entire batch. This would have the advantage of the largest possible baseline and thus allow the determination of the true mean rather than a sample mean. However, it would also require a significant change in ballot printer behavior and so is not incrementally deployable. For the moment, therefore, we must assume that election officials will be responsible for estimating μ.

Election officials might also determine $\hat{\mu}$ immediately prior to the election. Intuitively this seems better than afterwards: the ballots are all assembled in a single location and packaged in well-defined units with known counts, so it seems easy to weigh a large number of them (recall that the accuracy of this measurement scales with the square root of the number of ballots measured.) In principle, it might be possible to weigh all the ballots in a precinct or county before they are distributed without undue effort. Unfortunately, in many cases election officials are not dealing with simple stacks of ballots but rather ballots in boxes, wrapped in shrink wrap, attached to pads, etc. The weight of the packaging must then be determined and subtracted, which seems problematic. It would of course be possible to select some number of ballots for the purposes of calibration and remove the packaging, but it's not clear that those ballots could then be returned to circulation without unduly affecting local procedures. This is especially true in counties where the ballots are distributed on pads. Accordingly, we believe it is currently most practical to determine $\hat{\mu}$ after the election but prior to the audit.

[8] Note that gravitational variation is sufficiently large that it is very important to calibrate the scale to local gravity if μ is to be determined in a separate location from where the audit is performed.