Process: Election Audit

Issue Date: August 9, 2010

Owner: Elections Coordinator – Carlos Webb

Overview

The Election Audit is a quality assurance procedure that checks the accuracy of the machine tally of ballots (Ref 6). Ballots for a contest are sampled, hand counted and the results checked against the vote tabulation. In the risk-minimizing approach that we use, the sample is specifically constructed so as to minimize the risk of confirming an incorrect election outcome (Ref 7).

The Election Audit is one of the two quality assurance procedures we use when certifying the election. The other is the Election Reconcile which investigates the conduct of the election. It compares the number of voters, the ballot inventory, the ballots voted and the vote credit assigned. The results of both of these procedures are primary inputs to the Election Canvass.

We are fortunate that we can piggyback on an excellent audit strategy employed in 2008 (Ref 1). Because of that effort we have some experience in statistical audit techniques that form the foundation for this audit process.

Audit Requirements - Colorado Law & SOS Rules

The audit must conform to current state statute and SOS rules:

- CRS 1-7.514 (Random Audit) manual audit of 5% of voting devices or, for central count counties, a "specified" percentage of ballots counted in the county. Or, use any method that provides a higher level of statistical confidence.
- SOS Rule 11.5.4 (Post Election Audit for Central Count counties) random selection of one Central Count scanner (all races on no more than 500 ballots) and 5% of DREs (all races on all of the paper trails).

Boulder County will use a risk-minimizing audit (Ref 7) -- an option that meets the "higher level of statistical confidence" part of CRS 1-7.514. Risk-minimizing audits typically require limited resources for races with wide margins of victory while investing greater resources in close races. The technique increases coverage and reduces expense.

State law also provides provisions for an automatic recount (CRS 1-10.5-101). Since the need for a recount is only determined after the election is certified, there are no direct interactions between the audit and the recount. However, if a contest is likely to be included in a recount, to minimize risk to the paper ballots and VVPAT paper trails, we will postpone any audit of the contest until after the recount.

Audit Procedure Highlights

At a high level, the audit procedure is straightforward: select some contests to audit, select groups of ballots on which to hand count those contests, compare the hand count with the machine count of those same ballots.

Selecting Contests

When audit resources are limited, it's useful to statistically avoid selecting contests that have wide margins or are uncontested. This focuses the audit on the contests where the outcome may be in question. As was done in the 2008 General Election (Ref 1) and in the 2009 Coordinated Election (Ref 9), we can do this by selecting contests with a probability proportional to the inverse of the contests' margin. Close contests, those with a small margin, are more likely to be selected.

We don't require that all contests be selected in this manner. For example, we may be mandated by the SOS to audit a particular contest. Or, it could be that there is a national or statewide contest that is particularly heated and public interest suggests that the contest should be audited. However, the majority of contests need to be selected statistically, as above, since there is higher confidence in the election when no contest is immune from being audited.

Selecting Ballots (selecting audit units)

Audit units are groups of ballots that are selected for hand counting a contest. Audit units, in general, must adhere to two requirements: (1) that the ballots can be retrieved as a group and be available for the hand-counted, and (2) that a machine count of the group is available for comparison.

Because election results are eventually reported by precinct, it would be nice if the audit unit could be all of the ballots associated with a precinct. However in our central count system, ballots are grouped as they arrive back in the Elections Department and are not resorted into precincts. It would be difficult to use precinct ballots as the audit unit.

In Boulder, voted ballots that arrive back in the Elections Department are grouped, processed (i.e., scanned and resolved), aggregated with other groups and counted. These aggregates are called MBBs and they have exactly the characteristics we need in an audit unit – groups of ballots, archived together and with a machine count[1]. In the 2008 election there were over 525 MBBs, each containing between 1 and 1760 ballots. The audit consisted of randomly identifying a set of MBBs, hand counting them and comparing with the MBB's tabulated results. The 2008 audit hand counted select contests on 24 MBBs for a total of 69 MBB-contest pairs. In 2009 a similar methodology picked 21815 votes from 20 contests and 78 MBBs out of a total of 425 MBBs.

So, how many audit units do you randomly select to have a statistically valid audit? You could select a fixed percentage of the audit units. California uses this approach and audits all contests on a fixed 1% of the ballots. Or, you could have a tiered approach like what was proposed in HR 811: (1) if vote margin is less than 1% then audit 10% of the audit units, (2) if the margin is more than 1% but less than 2% then audit 5%, (3) otherwise audit a fixed 3% of the audit units. We choose to use a risk-minimizing statistical approach (Ref 3, 4, 6, 7) because of its improved efficiency. This approach focuses on the fewest number of audit units to achieve a specified statistical confidence, or probability, that an outcome-affecting error would be detected. This choice gives us an opportunity to audit more contests for the same audit cost.

Our audit unit selection relies on the NegExp algorithm detailed in Reference 4. It uses the contest's margin, the size (number of ballots) of the audit unit and a parameter that define the rigor of the algorithm to select audit units. The algorithm is well suited to large, complex elections in that selected audit units can be used across several contests, thus minimizing some of the administrative overhead of running an audit. This is an active area of research and in the two years since NegExp has been published, more robust algorithms have been developed and are being investigated for future audits.

Comparing Results

Once the hand count of the selected MBBs for a contest is completed, the results are compared to the machine count of the same ballots. If the difference between the hand count and the machine count differ by ⌈0.5%⌉ or more of the ballots hand counted, the audit is escalated. This "escalation trigger" prescribes a sequence of troubleshooting activities which help (1) to identify the source and extent of the problem with the contest and (2) whether other contests are impacted.

Confirm that there is a discrepancy

[1] It's time consuming to produce individual MBB results during the conduct of the election. Instead, we produce cumulative MBB results which are processed after the election to produce the individual MBB tally results.

- Verify that the MBB is complete. Confirm that the hand counters and the Ballot Now and Tally systems saw the same number of ballots.
- Re-verify the hand count, perhaps using a different hand count team.

Look for "unusual" ballots as source of discrepancy

- Manually inspect the ballots. Identify any that are "unusual" – ie, have stray marks, or are incorrectly marked, or are inconsistently marked -- and compare how the election night Resolution Team resolved them. If the discrepancy is due to a difference of opinion between the hand count team and the election night Resolution Team, document the difference and de-escalate.
- Visually inspect the scanned images. Using the Ballot Now Resolution Screen, compare image quality and resolution of the contest on all ballots on the MBB. Look carefully at ballot fold lines. If the Ballot Now and hand count resolutions agree, yet the tally of the MBB doesn't, investigate Tally as the source of the discrepancy and proceed to "Investigating Tally Discrepancies".

Investigate resolution differences

- Folding in the results of other MBB-Contest audits, look for trends across scanners and contests. Consider re-scanning the MBB using a different scanner. If this resolves the discrepancy then investigate how the original scanner was used during the election. Investigate any discrepancies in the MBBs that have already been audited and that used this scanner. Decide if non-audited MBBs from this scanner need to be re-scanned.
- If the discrepancies seem to relate to this particular contest and the discrepancies cannot be resolved by re-scanning on different scanners, increase the number of MBBs audited for this contest. If needed, this escalation may result in a complete handcount of the contest across all MBBs. If this doesn't resolve the discrepancy (i.e., total discrepancies be within ⌈0.5%⌉) then consider reinstalling the Ballot Now Trusted Build, re-scanning and re-tallying.

Investigate Tally Differences

- When the election night resolutions and the hand count resolutions are identical yet Tally shows different election results, you have to suspect Tally or the MBB. Inspect the MBB to see if there was a fault writing to it. In diagnosing this, see if Tally can "recover" the MBB.
- As a last resort, securely archive the Tally hardware and software for later scrutiny and, with SOS support, re-run the tally of all MBBs on a different, trusted Tally system. Compare the election results from the trusted Tally against the hand counts.

Audit Schedule

The schedule for the audit is tight. Ideally, it would be best to wait as late as possible after Election Day so that all ballots are considered in the audit. However, audit results are a major input to the Canvass Board, which is required to certify the election within 13 calendar days of a Primary (or 18 days for other elections). With those two constraints in mind, the audit fits into a narrow window that starts on the Friday following the election. With this early snapshot of the election, it is likely that some small portion of Provisionals and mail-in cures will be missed. The schedule for the 2010 Primary is as follows:

Week 1	Tuesday	Election Day
	Late Thursday (8/12)	Take snapshot of Score and Tally as the basis of the Election Audit and the Canvass.
	Late Friday (8/13)	Publish Audit Setup information.
	Saturday (8/14)	Meet with Audit Board to select audit parameters, select contests, select MBBs.
Week 2	Early Monday (8/16)	Make available selected MBBs for hand counting.
	Mon-Fri (8/16-8/20)	Hand count the selected MBBs – Monday noon through Friday.
	Late Thurs (8/19)	Canvass Board begins assessment with non-audit aspects of the election.
	Late Sunday (8/22)	Publish Audit Report.
Week 3	Early Monday (8/23)	Canvass wrapup..
	Mon EOD (8/23)	Certify election.

Audit Budget

The audit budget covers the hand count and escalation of an estimated 3% of the expected ballot-contests in the election. As contests are selected, their ballot-contest estimates will be deducted from the budget. Contests will be selected as long as there are resources left in the budget to audit them.

If the auditing of randomly selected contests concludes before the budgeted time is up, we will accept nominations from the community for additional MBB-contest pairs to audit. The selections will be made by the audit team, as time permits.

Acknowledgments

This plan is a derivative of the Boulder County 2008 General Election audit that was lead by Neal McBurnett. Neal's subsequent discussions and comments on drafts of this plan have been greatly appreciated.

References/Resources

1. Boulder County 2008 General Election Audit, Neal McBurnett, http://bcn.boulder.co.us/~neal/elections/boulder-audit-08-11.

2. In Consideration of Ohio Audits in 2008, Feb 19, 2008, Ohio Joint Audit Working Group, http://www.caseohio.org/Documents/Reports/Ohio_Audit_White_Paper_Feb_2008.pdf.

3. Recommended Audit Procedures, June 2, 2008, Ohio Joint Audit Working Group, http://www.caseohio.org/PageDetails/Audits/Ohio_Recommended_Audit_Procedures.pdf.

4. On Auditing Elections When Precincts Have Different Sizes, Aslam, Popa, Rivest, Proceedings of the 2008 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT). July 2008, http://www.usenix.org/event/evt08/tech/full_papers/aslam/aslam_html. See also, slides at http://www.usenix.org/event/evt08/tech/popa-slides.pdf.

5. Checking Election Outcome Accuracy. Post-Election Audit Sampling, Kathy Dopp, July 18, 2009, http://kathydopp.com/serendipity/index.php?/archives/44-RELEASE-New-improved-methods-for-post-election-audit-sampling.html.

6. Implementing Risk-Limiting Post-Election Audits in California, Hall, et al, August 10-11, 2009, http://www.usenix.org/event/evtwote09/tech/full_papers/hall.pdf.

7. Best Practices: Risk Limiting Audits, ElectionAudits.org, http://www.electionaudits.org/bp-risklimiting.

8. A "Sum of Square Roots" (SSR) Pseudorandom Sampling Method For Election Audits, Ronald Rivest, April 25, 2008, http://people.csail.mit.edu/rivest/Rivest-ASumOfSquareRootsSSRPseudorandomSamplingMethodForElectionAudits.pdf.

9. Post-Election Audit Report, 2009 Coordinated Election – Boulder County, November 29, 2009, http://www.bouldercounty.org/clerk/elections/Audit%20Files/2009%20Audit%20Report.pdf.

Inputs

1. MBB Cumulative Reports from Hart Tally.
2. Contest margins when contest extends beyond the county.

Outputs

1. Election Audit report

Change History

Month Day, Year i.e. (March 27, 2009)	Description of change i.e. (Initial Issue)
August XX, 2009	Initial draft
September XX, 2009	Second draft
October 21, 2009	Version 1.0
August 9, 2010	Version 2.0 for 2010 Primary

Election Audit Process Details

A trustworthy audit requires transparency. The audit plan and the uncertified election results (per audit unit) need to be published and available before the audit is conducted. Software tools need to be transparent as well; the use of tools to expedite portions of the audit are acceptable but auditors need to be able to manually verity the output generated by any tool. While this is particularly true for third-party software, even use of off-the-shelf office software like spreadsheets needs to be open and review-able.

Verification steps are built into the audit details described below. How extensive these steps are pursued is left up to the discretion of the audit team.

Audit Setup

1. [Elections Staff] Construct Individual MBBs from Hart Cumulative MBBs

The statistical algorithms used in the audit require election results by MBB. However, individual MBB reports are time consuming to produce and impractical to get while ballots are being counted. But cumulative MBB reports are available. A cumulative MBB report contains the MBB's election results summed with all previously tallied MBBs. Individual MBBs reports can be constructed by subtracting the successive cumulative reports.

The timing of when to start the audit may be an issue. On one hand, we would like to wait until all ballots are processed and are accounted for in an MBB. However, the results of the audit are a major input into the election certification process, which statutorily has a short fuse. This means that the audit may have to proceed before all ballots are processed. In the past this meant that Provisional Ballots weren't included.

The first step in setting up the audit is to create Individual MBB reports from Cumulative MBB reports, using some third party software tools. The resulting Individual MBBs will be verified for correctness when the audit team is assembled.

a) Copy the Hart Tally directory containing the MBB cumulative xml reports. This directory should contain pdf and xls/csv files as well as a directory containing an xml for each cumulative report.

b) Execute the perl script "SimplifyCumulativeMBBs.pl". This program creates a simplified, uniform version of the more complicated xml produced by Tally. Two parameters are required – the input directory created above, and an output directory where the simplified, cumulative MBBs will be written.

c) Execute the perl script "SpotMissingFiles.pl", redirecting output to "MissingFileCheck.csv". Given a directory containing the "raw" Tally cumulative reports (as in Step 1), this program generates a report that is useful to determine if any xml files are missing. Generating the Tally cumulative reports are tedious and, consequently, error prone. Sometimes pdf or xls versions of a cumulative report will be saved but the xml save is missed. The SpotMissingFiles.pl program summarizes the files in the directory in a way that highlights when an xml file is missing.

d) Determine the order that the cumulative MBBs were tallied by executing the perl script "CumulativeSortOrder.pl" and redirecting output to "SortOrder.csv". This program inspects each simplified cumulative MBB and outputs relevant statistics on each. In particular, there are three fields associated with each cumulative MBB that should be reviewed: MBB sequence number, cumulative file creation time and cumulative cast votes. Open SortOrder.csv as a spreadsheet and, using these three fields, re-sort it so that the order reflects the order that the MBBs were read into Tally. Once resorted, save the file as a csv.

e) Create individual xml MBBs by deconstructing the cumulative xml MBBs according to the tally order determined above. Execute the perl script "CreateIndividualMBBs.pl" with the following command line parameters (1) the tally order csv file created above, (2) the directory containing the simplified cumulative xml MBBs and (3) an output directory where the individual xml MBBs will be written.

   Note: if several Tally databases were used, you will have to divide the csv file by database and execute each part separately.

f) To test the integrity of the individual MBBs, execute the perl script "RecreateCumulativeMBB.pl" with command line parameters (1) the tally order csv file created above, (2) the directory containing the individual MBBs and (3) a new output directory where the recreated cumulative MBBs will be written. Verify that the last recreated cumulative MBB agrees with the results captured in the original Tally cumulative pdf.

2. [Elections Staff] Determine In-County Voters per Contest

The number of voters per contest is equal to the number of ballots received – usually this is the number cast votes plus overvotes plus undervotes. Calculate this for each contest on each MBB by executing the perl script "VoterCounts.pl". Pass as a parameter the directory containing the Individual MBBs and redirect output to "VoterCounts.csv". The csv file contains the number of ballots or voters per MBB per contest as well as the county's grand total margin of victory between the top two candidates.

3. [Election Staff] Protect Voter Privacy in Small DRE MBBs

When reporting election results we protect the privacy of voters by ensuring that the precinct reports don't include small sample sizes. Results showing 25 or fewer voters are combined protect the secrecy of each individual's vote.

In VoterCounts.csv, manually combine MBBs when the number of voters for a countywide contest is less than 25.

4. [Election Staff] Adjust Margins for Multi-County Contests

The risk-minimizing algorithms we use require an absolute margin, not a countywide margin, between the winner and loser. For multi-county contests, find the absolute margins and total voter counts and adjust VoterCounts.csv accordingly.

5. [Elections Staff] Determine Contest Selection Probabilities

Efficient audits statistically avoid auditing contests that have wide margins. This is done by selecting contests with a probability proportional to the inverse their margin. In this step, probabilities are assigned to each contest so that when the audit is conducted, dice can be used to select contests.

a) For each contest, determine the Fractional Margin m=M/V where M is the absolute margin and V is the contest's total voters.

b) Determine the Inverse Fractional Margin 1/m. If the Fractional Margin is zero, assign an arbitrary large number to the Inverse Fractional Margin.

c) Determine Contest Selection probability as the proportion of this contest's Inverse Fractional Margin to the total (i.e., all contests' Inverse Fractional Margins).

d) To facilitate a random selection using dice, iteratively sum the probabilities to construct a running cumulative probability for each contest.

6. [Elections Staff] Determine MBB Selection Probabilities

After a contest is selected, MBBs for hand count need to be selected. The selection is probabilistic, with probabilities assigned by an algorithm that depends on the number of voters per MBB (Step 3, above), the absolute margins (Step 4, above) and the statistical confidence desired in the algorithm. Higher confidence translates higher likelihood that an error, if present, will be detected.

We will use the NegExp algorithm described in detail in Reference 4 to calculate these probabilities. A detailed summary is contained in Attachment 1. This algorithm is flexible and is our algorithm of choice for large elections. For transparency the NegExp algorithm will be implemented in a spreadsheet so that the audit team can inspect formulas and test the implementation.

7. [Election Staff] Generate Pseudorandom Numbers Using SSR Algorithm

The algorithm for selecting MBBs requires random numbers and the above probability tables. Using the SSR pseudorandom algorithm of Reference 8, provide a spreadsheet that takes a 15-digit seed and produces a large number of pseudorandom numbers.

8. [Elections Staff] Publish Audit Details

For transparency, the results of the above audit setup will be published on the county website so that the public can confirm the work done so far. Include the following information along with SHA1 checksums (http://support.microsoft.com/kb/841290):

- VoterCounts.csv
- Margin adjustments for multi-county contests
- Contest Selection probabilities
- MBB selection probabilities for NegExp
- SSR pseudorandom number generator
- A zip containing the Hart Cumulative MBB pdf and xml files
- A zip containing the Individual MBBs xml files produced in Step 1, above.
- This Audit Procedure document

Conducting the Audit

9. [Elections Staff] Convene the Audit Team

Convene the audit team and have them sign an affidavit stating that they will honor the privacy of the ballot and protect the materials they'll be working with. Give them an overview of the process. Emphasize that they will be working in pairs from different political parties and will need to verify each other's work.

10. [Election Staff and Audit Team] Confirm the Generation of Individual MBBs

If necessary download the audit setup information published in Step 8. Recalculate the SHA1 checksum and verify that the checksums haven't changed.

Test the integrity of the tools used to create the individual MBBs. Have audit team members select several individual MBBs at random and a contest. Provide them with (1) Tally's cumulative pdf report for this MBB, (2) the cumulative pdf report from the preceding MBB, and (3) a print of the individual MBB xml. The audit team should verify that the individual MBB is correct for the contest(s) selected.

11. [Elections Coordinator and Audit Team] Establish MBB Selection Parameters

For each contest, determine the level of confidence desired for each contest. Audit confidence, in this context, is loosely defined as the likelihood that the audit will be strong enough to detect an results-impacting error in the contest if one exists. This parameter along with contest margin determines the work load of the audit – specifically how many ballots will be hand counted.

In 2008 we picked 99% confidence for state contests and 75% for local contests. In 2009 we raised the local contests to 80%. It's useful to plug these numbers into the MBB selection spreadsheet (Step 10, above) to get a feel for the impact that different confidence levels can make.

12. [Elections Coordinator and Audit Team] Select Contests to Audit

First, determine if the SOS has mandated that certain contests be audited. Then, eliminate uncontested contests and contests that are targeted by the SOS for a re-count.

Select additional contests randomly, as follows:

- Construct a random number x, where 0 <= x < 1. The number can be the result of rolling 4 die. Or, the audit team may decide to just grab a block of pseudorandom numbers that were constructed in Step 7.
- Using x, scan the ordered Contest Selection probabilities table, selecting the contest having a cumulative probability immediately greater than the value x.

For each contest selected, estimate the workload on the audit team. Workload is the number of ballots that the team will scrutinize for each contest. It's pretty easy to estimate workload using the NegExp algorithm. See Reference 4 for details.

13. [Elections Staff] Prepare Hand Count Tally Sheets

For each contest selected, prepare tally worksheets for the hand count.

14. [Elections Staff and Audit Team] Select MBBs for Each Contest

The NegExp algorithm has a nice property that MBB probabilities can be combined for the selected contests, allowing the MBB selection to cover all selected contests. To combine NegExp probabilities so that a single set of MBBs can be used to audit several contests:

- The combined NegExp probability for an MBB is the maximum of the NegExp probability for each selected contest.
- Do this for each MBB.

Select MBBs for NegExp (NegExp #6 in Attachment 1):

- Use four dice or the pseudorandom numbers created in Step 7 to construct a random number x, where 0 <= x < 1.
- If x is less than the combined NegExp probability for the MBB, select the MBB for auditing.
- Do this for each MBB.

With the number of MBBs approaching a thousand, NegExp will require a lot of random dice throws. Alternatively, the audit team may choose to use the SSR random number table saved in Step 7. If this is the case, the team should use the dice to construct a single random 15-digit seed which then randomly select the MBBs.

15. [Audit Team] Audit the MBBs

For each selected batch, an individual hand count is performed for the selected contests. Hand counts use 4-person teams. One person (the Caller) reads the ballot for the target contest and calls out the vote. A second person (the Witness) observes the Caller for accuracy. Two people (Tallier #1 and Tallier #2) separately record the vote. The talliers compare their results periodically and investigate if there are discrepancies.

It is important that the counting be done "blind". But since the Individual MBB reports are published before the counting begins, this is somewhat problematic. Where possible, the MBBs will be split between two counting teams so that no one team will be responsible for the entire MBB hand count.

16. [Election Staff and Audit Team] Compare the Hand Count and the Machine Count

When contest hand counting is completed, the elections staff will collect the hand count results in a spreadsheet and compare to the individual MBB tallies.

We are executing the audit to detect errors that may impact the election outcome. If any of the numbers differ by ⌈0.5%⌉ or more of the ballots cast for the contest on the MBB, then the hand count for the MBB-Contest is escalated through a series of troubleshooting steps until the discrepancy is resolved or the scope of the problem is understood and fixed:

Confirm that there is a discrepancy

- Verify that the MBB is complete. Confirm that the hand counters and the Ballot Now and Tally systems saw the same number of ballots.
- Re-verify the hand count, perhaps using a different hand count team.

Look for "unusual" ballots as source of discrepancy

- Manually inspect the ballots. Identify any that are "unusual" and compare how the election night Resolution Team resolved them. If the discrepancy is due to a difference of opinion between the hand count team and the election night Resolution Team, document the difference and de-escalate.
- Visually inspect the scanned images. Using the Ballot Now Resolution Screen, compare image quality and resolution of the contest on all ballots on the MBB. Look carefully at ballot fold lines. If the election night and hand count resolutions agree, yet the tally of the MBB doesn't, investigate Tally as the source of the discrepancy and proceed to "Investigating Tally Discrepancies".

Investigate resolution differences

- Folding in the results of other MBB-Contest audits, look for trends across scanners and contests. Consider re-scanning the MBB using a different scanner. If this resolves the discrepancy then investigate how the original scanner was used during the election. Investigate any discrepancies in the MBBs that have already been audited and that used this scanner. Decide if non-audited MBBs from this scanner need to be re-scanned.

- If the discrepancies seem to relate to this particular contest and the discrepancies cannot be resolved by re-scanning on different scanners, increase the number of MBBs audited for this contest. If needed, this escalation may result in a complete handcount of the contest across all MBBs. If this doesn't resolve the discrepancy (i.e., total discrepancies be within ⌈0.5%⌉) then consider reinstalling the Ballot Now Trusted Build, re-scanning and re-tallying.

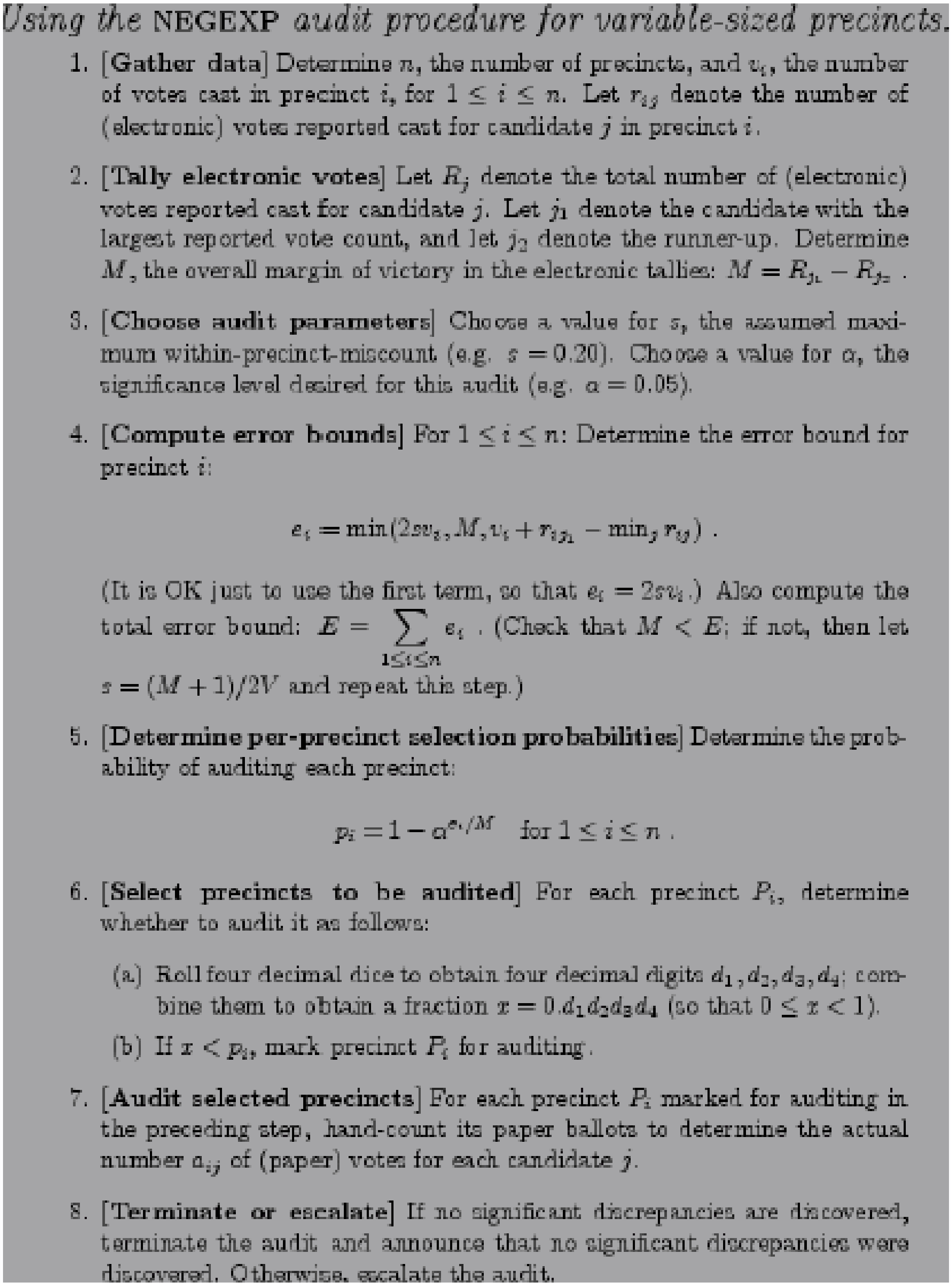
<u>Investigate Tally Differences</u>

- When the election night resolutions and the hand count resolutions are identical yet Tally shows different election results, you have to suspect Tally or the MBB. Inspect the MBB to see if there was a fault writing to it. In diagnosing this, see if Tally can "recover" the MBB.
- As a last resort, securely archive the Tally hardware and software for later scrutiny and, with SOS support, re-run the tally of all MBBs on a different, trusted Tally system. Compare the election results from the trusted Tally against the hand counts.

17. [Election Staff] Audit Report

The election staff will prepare the audit report and publish it to the public.

Attachment 1 – Detailed Procedures for the NegExp and PPEBWR Algorithms[2]



Using the NEGEXP *audit procedure for variable-sized precincts.*

1. **[Gather data]** Determine n, the number of precincts, and v_i, the number of votes cast in precinct i, for $1 \le i \le n$. Let r_{ij} denote the number of (electronic) votes reported cast for candidate j in precinct i.

2. **[Tally electronic votes]** Let R_j denote the total number of (electronic) votes reported cast for candidate j. Let j_1 denote the candidate with the largest reported vote count, and let j_2 denote the runner-up. Determine M, the overall margin of victory in the electronic tallies: $M = R_{j_1} - R_{j_2}$.

3. **[Choose audit parameters]** Choose a value for s, the assumed maximum within-precinct-miscount (e.g. $s = 0.20$). Choose a value for α, the significance level desired for this audit (e.g. $\alpha = 0.05$).

4. **[Compute error bounds]** For $1 \le i \le n$: Determine the error bound for precinct i:

$$e_i = \min(2sv_i, M, v_i + r_{ij_1} - \min_j r_{ij}) \ .$$

   (It is OK just to use the first term, so that $e_i = 2sv_i$.) Also compute the total error bound: $E = \sum_{1 \le i \le n} e_i$. (Check that $M < E$; if not, then let $s = (M+1)/2V$ and repeat this step.)

5. **[Determine per-precinct selection probabilities]** Determine the probability of auditing each precinct:

$$p_i = 1 - \alpha^{e_i/M} \quad \text{for } 1 \le i \le n \ .$$

6. **[Select precincts to be audited]** For each precinct P_i, determine whether to audit it as follows:

   (a) Roll four decimal dice to obtain four decimal digits d_1, d_2, d_3, d_4; combine them to obtain a fraction $x = 0.d_1d_2d_3d_4$ (so that $0 \le x < 1$).

   (b) If $x < p_i$, mark precinct P_i for auditing.

7. **[Audit selected precincts]** For each precinct P_i marked for auditing in the preceding step, hand-count its paper ballots to determine the actual number a_{ij} of (paper) votes for each candidate j.

8. **[Terminate or escalate]** If no significant discrepancies are discovered, terminate the audit and announce that no significant discrepancies were discovered. Otherwise, escalate the audit.

[2] Lifted from "On Auditing Elections When Precincts Have Different Sizes", Aslam, Popa, Rivest.