Obtaining Batch Reports for Audits from Election Management Systems: ElectionAudits and the Boulder 2008 Election

Neal McBurnett, Boulder CO, neal@mcburnett.org, http://neal.mcburnett.org/

Abstract

Obtaining the data necessary to do the most effective and efficient audits from current Election Management Systems (EMS) can be difficult. For the Boulder 2008 election we developed procedures and software (ElectionAudits) to obtain suitable vote counts from the Hart InterCivic EMS, and automated many of the steps necessary to analyze the data and run the audit . Since Boulder stores its ballots by batch rather than by precinct, we also had to generate a vote count report by batch. We did this by using the EMS to report overall vote counts for each candidate after each batch was processed. ElectionAudits could then subtract each count from the previous count to produce vote counts for the batch. The ability to report vote count results by batch rather than by precinct can dramatically improve the efficiency of the audit if the batches are small, and would be a helpful feature in an EMS. In order to enhance the transparency and usability of the audit, it would also be helpful to report what was audited, information on discrepancies, and audit results in a standard format. An extention to EML for audit results would be appropriate.

Introduction

An auditable election system and a good audit are essential parts of achieving confidence in the integrity of an election, and "software independence" [Rivest 2006] is an important goal for elections administrators.

With currently deployed Election Management Systems ("EMS"), post-election vote tabulation auditing requires the use of paper ballots, or when votes are stored electronically, a Voter Verified Paper Audit Trail. The EMS generates an auditable report, with vote count subtotals broken down by audit unit (precinct, machine or batch). The audit involves generating a sample of the vote counts reported by the system, and comparing them to hand counts of the corresponding paper records.

Besides the overall totals used to decide who won an election, EMS's generally produce detailed results broken down by precinct. Precinct reporting is important to many election observers and to political parties. In the past it has also been suitable for auditing purposes, since in the past nearly all ballots were cast in precincts on election day. It was thus easy to hand-count all the votes for a given contest in a given precinct, and compare the hand counts with vote counts from the precinct reports.

Today, however, many election jurisdictions do not organize their ballots strictly by precinct, making it hard to access all the ballots for those precincts selected for the audit. For example, mail-in ballots may remain in batches formed as they are received. Also, in many jurisdictions, some voters cast early votes on DRE machines at a central location, so that the Voter Verifiable Paper Audit Trail (VVPAT) for the machine has ballots from many precincts on a single long roll of paper which can be problematic to split up into pieces by precinct.

Auditing practices have also evolved significantly in recent years[BestPractices2008][LWV2009]. The most effective and efficient audits are now complex enough that specialized auditing software is very helpful. Auditing software can statistically analyze the vote counts and help randomly select audit units with probability propotional to size, which can sometimes reduce the number of ballots counted by a factor of 2 while retaining the same probability of finding a problem[Aslam2008]. It can also analyze discrepancies and determine whether enough has been audited or whether additional units must be selected in order to meet the confidence goals of the audit.

We present here our experiences related to formats and data management in the audit of the Boulder County General Election in 2008. The main problem was a mismatch between the organization of the election reports and the organization of the physical storage of the paper ballots. Additional problems included non-standard, difficult-to-parse report data and omissions of important data in most of the reports. We resolved the problems successfully, and present the procedures and the open source software used to address the problems and the lessons we learned. In the process we enumerate our data requirements for modern post-election tabulation audits.

Auditing data requirements

It is very important for the transparency of an audit to publish detailed audit reports, with full election results on each audit unit (a batch, precinct or DRE machine), before the audit or random selection is done.[BestPractices2008]

The data for each audit unit must include at least this information:

1. audit unit identifier and tracking information sufficient to locate the paper ballots
2. contest name
3. total number of ballots, including those on which the contest does not appear, for ballot reconscillation
4. number of votes for each candidate
5. number of undervotes (no candidate selected)
6. number of overvotes or other spoiled ballots

The number of candidate votes, undervotes and overvotes must sum to the number of ballots on which the contest appears.

To do an efficient state-wide audit with many modern auditing approaches, detailed contest-wide election data from all jurisdictions needs to be imported and parsed and analyzed before the audit can begin. One reason for this is the need to calculate the overall margin, which is used to determine how many audit units need to be initially selected. For the most efficient approaches such as those described by [Aslam2008] and [Hall2009] you also need to analyze the sizes of all the audit units in the contest and the vote counts in each audit unit.

Selection of which contests to audit may also require analyzing election result data. In our Boulder audit the contest selection algorithm included as many contests as possible within a predefined auditing budget. The budget specified how many audit units we expected to audit, assuming that no major discrepancies were found.. The contests were selected randomly, with probability proportional to the inverse of the margin of victory. As each contest was selected, the expected number of audit units to audit was also calculated. Additional contests were then selected for the audit until the budget was used up. The contest selection thus depended on the margins from data on all contests. In the future it may also take into account the undervote rate of each contest (as a measure of public interest in the contest).

Auditors and the general public also need easily parsable data in order to analyze the audit units for suspicious patterns and target interesting units for selection.

Given the many circumstances under which auditable vote count data is helpful, it is important to be able to easily obtain it from EMS's. Unfortunately this is often not the case, as Hall et al. found in four California counties[Hall2009] and as we found in Boulder County, Colorado.

Obtaining Batch Results for the Boulder Audit

For the 2008 General Election, Boulder County enhanced many auditing practices, after obtaining approval from the office of the Colorado Secretary of State.

Boulder County received about 170000 ballots, and about 70% were mail-in ballots. There were an average of about 50 contests per ballot and most ballots too two sides of two sheets of paper. The ballots were marked with barcodes to indicate the precinct, so the EMS can report by precinct after the election. But.like most of Colorado, the county does not sort incoming ballots into piles by precinct. They remain in the batches formed during the initial processing.

For the audit, we couldn't use the precinct report since it would be too hard to find all the ballots for any given precinct. So we created a batch report using new procedures and new software - the open source ElectionAudits software.

The new procedures involved generating a full “cumulative” election report with the Hart Tally system after processing each batch of ballots. This is the same sort of report that is generated to publish up-to-date results during election night, but done more frequently. For this election it was done 525 times during the original tallying process,

as each “Mobile Ballot Box” memory card (MBB) was fed into the tally system. Most MBBs contained information from ballots scanned on the Hard BallotNow system, but some contained information from ballots cast on Hart's eSlate DREs.

Hart's tally system can produce reports in many formats. They appear to use the Crystal Reports software to generate all of them. Most of the formats are very hard to process by machine. For example we hoped that the CSV (“comma-separated-values”) report would be a simple table with a row for each vote count and consistent columns presenting the audit unit id, contest, candidate, vote count, etc. But instead there were no column headings and each row was a mysterious sequence of numbers, names, and percentages. The columns were not consistent from row to row. It is not at all clear how to parse each row.

Most other report formats suffered similar problems. Many also did not have critical information like the number of undervotes or overvotes for a given contest.

The best Hart report format for our purpose was their XML report. Unfortunately they were not using the Election Markup Language (EML) format standard for elections. It was also not in Hart's own “EDX” XML format for elections – it appears that that is not supported in the version of the Hart software that Boulder had (System Version 6.0). Instead it was a custom Crystal Reports XML format.

It had a lot of extraneous Crystal Reports information which seemed related to the way the PDF was laid out, but the elements were consistently labelled so they could be parsed. The only thing lacking from the Hart XML report was the cumulative number of ballots read in to date. This is important information for tracking ballots. In many elections it can be inferred from the total of all vote counts for a contest that everyone is eligible for. But in Boulder's case, even the total of all votes in the presidential race couldn't be used as a ballot count. Some voters are people who live in another county or state and are only eligible to vote in local tax districts on property tax elections. So we had to use other means to get the cumulative ballots counts. We ended up finding it in the pdf reports, and extracted it using third-party pdf processing tools and custom pattern matching scripts.

ElectionAudits read in the cumulative XML reports, and generated vote counts by batch for each candidate by subtracting each vote count for each candidate in each cumulative report from the corresponding vote count in the previous cumulative report.

Then we published that data on our web site for public transparency.

Audit Selections and Other Procedures

See the Boulder election web site for further information on the procedures, the publically verifiable method of generating random numbers, our contest selection approach, the NEGEXP method for selecting audit units for each contest, etc.

http://bcn.boulder.co.us/~neal/elections/boulder-audit-08-11/

ElectionAudits Software

In order to facilitate future audits both in Boulder and beyond, we encapsulated much of the process in a new software application named ElectionAudits. It is open source software, offered under the very liberal “MIT” license, and is suitable for use by itself . It could also be incorporated into existing or future EMS's.

ElectionAudits helps with several facets of the task of auditing elections:

* Imports standard election report files, currently including Hart's XML and a simple CSV format.
* Protects voter anonymity by combining results for very small precincts or batches into larger audit units.
* Works with election systems which can't produce batch results by themselves, and in jurisdictions that don't sort all ballots by precinct
* Facilitates risk-limiting audits by calculating relevant statistics based on the margin, number, size, and results of the audit units. Support for various selection procedures, including NEGEXP and many aspects of PPEBWR and SAFE, is provided, with more on the drawing board.
* Implements the "Sum of Square Roots" (SSR) Pseudorandom Sampling Method[Rivest2008], which allows the entire set of selections for both contests and audit units with a contest to be automated based on a truly random seed value generated from dice throws. The selections are easy for the public to verify.

* Provides a convenient platform for publication and archiving of auditable reports for the public.
* Provides an ad-hoc query calculator form lets you quickly enter information about hypothetical contests and get selection statistics for them also. This supports planning for future audits, e.g. deciding how many batches to report on so as to minimize the fraction of the ballots you'll have to audit.

The software is writtin the Python programming language, using the Django framework for both publishing on the web and for the user interface.

In the future we plan to add the ability to both read and produce election data in the EML format, which appears to have all the information we need in a much easier-to-parse format than the existing formats we've seen. We also plan to add other formats, and are already looking at the Sequoia reports which seem easy to add.

More information is available at https://launchpad.net/electionaudits

EML Enhancements

Because it is important to produce audit reports that election observers can analyze themselves, complete with selections, discrepancy information etc, we think it would be helpful to add another schema to EML for this purpose. Then we could also move to interoperability of auditing software, so that different jurisdictions in a state could report audit results in a way that allowed the Secretary of State to easily read in and analyze the data, and determine if escalation was necessary.

Lessons Learned

We learned that a "best practices" audit of multiple contests in a significant election can be done, with the proper data gathering techniques, advanced preparation, and specialized software.

The effort needed to generate the 525 cumulative EMS reports, for input into the batch report generation process, was time-consuming – it took on the order of 12 staff-hours of valuable time during the initial count of the election. We then spent over 3 days with over 10 people hand-counting ballots.[1] So given the higher efficiency of the NEGEXP selection method, the up-front batch reporting work allowed us to either save many more hours of time hand-counting enough ballots to get the level of confidence we got, or it allowed us to get a higher level of confidence than we would have gotten otherwise with the same amount of hand-counting.

It is *much* easier to audit effectively if results are reported in many small audit units. It is easier to get consistent hand counts on small units. Also, the statistical probabilities of finding problems are much more dependent on the number of audit units audited than on the total number of ballots, so auditing 65 small batches is just about as good as auditing 65 big batches, and much less time-consuming.

Of course, if the EMS was able to produce batch reports by itself, a great deal of time could be saved. This should not be difficult for most EMS systems we've seen, since they seem to keep track of batch information to some degree already in the internal databases, and it is not hard to generate the reports themselves. And given the advantages of auditing in small batches, it would be especially helpful if EMS's would support that without the additional work we had to do.

It is currently very hard in Colorado to get state-wide results for the contests with the necessary counts for under votes and over votes. We ended up relying on newspaper reports since this data is not gathered by the Secretary of State until after results are certified.

In the wake of the successful 2008 audit in Boulder, the state of Colorado changed the law in 2009 to move towards efficient risk-limiting audits over the next 5 years. We encourage the elections community to move towards ways to report election results in a standard format with small batches suitable for efficient auditing in order to make this process much easier.

---

1 In all we looked at about 13000 ballots, sometimes multiple times. In total there were about 40000 potential votes audited on those ballots, athough often the ballot did not contain the contest in question. We double- or triple-checked the counts until we got a consistent result, and used 2-person teams.

Acknowledgments

Credit for a multifaceted effort like this goes to many people: Boulder Clerk Hillary Hall for the leadership and resources, Crystal Christman and Scott Thomas for going beyond the call of duty, the members of the Canvass Board: Gary Boucher, Al Kolwicz, Deb Gardner and Carlos Webb for their assistance and ideas, and the hand counters and many others in the elections division. Much auditing experience came from Mark Halvorson and the members of the State Audit Working Group including Harvie Branscomb, Heleni Thayre and Pam Smith. Statistical expertise came from Ron Rivest, Raluca Popa, Javed Aslam, Arel Cordero, Kathy Dopp, Andy Bardwell, John McCarthy and the Thursday Auditing group members, including Philip Stark, Mark Lindeman, Lynn Garland, Eric Rescorla, Joseph Lorenzo Hall, Dennis Paull and Howard Stanislevic. Other contributors included Mary Eberle, Margit Johansson, Aaron D. Gerber, and Holly Lewis.
Responsibility for any problems with the audit design or implementation lies with Neal McBurnett.

References

- [Aslam2008] Aslam, Popa and Rivest. On Auditing Elections When Precincts Have Different Sizes, EVT 2008.
- [BestPractices2008] *Principles and Best Practices for Post-Election Audits*, http://electionaudits.org/principles/
- [Hall2009] Hall, et. al. *Implementing Risk-Limiting Audits in California,* EVT/WOTE 2009.
  [LWV2009] Election Audit Report, http://www.lwv.org/Content/ContentGroups/Membership/ProjectsTaskforces/Report_ElectionAudits.pdf 2009
- [Rivest2006] Ronald L. Rivest and John Wack. "*On the notion of ``software independence" in voting systems*", July 2006.
- [Rivest 2008] A "Sum of Square Roots" (SSR) Pseudorandom Sampling Method For Election Audits by Ronald L. Rivest.