RLA Software Requirements

April 13, 2017

Mark Lindeman, Neal McBurnett, John McCarthy, and RLA advocates

The purpose of this document is to summarize important requirements the Colorado Secretary of State's office should consider including in a Documented Quote and specification for software to help auditors carry out risk-limiting vote tabulation audits for Colorado's coordinated elections in November, 2017.

Overview

In order to carry out statewide risk-limiting vote tabulation audits (RLVTA) beginning in 2017, as mandated by law, the Colorado Secretary of State seeks software from qualified individuals and firms for a web-based RLVTA software system.

In general terms, the desired RLVTA system will help the Secretary of State's Elections Division to coordinate audit activities across Colorado's 64 counties:

1. upload ballot manifests and cast vote records from each county
2. retrieve contest results from the Election Night Reporting System
3. configure RLVTA parameters such as risk limits for different categories of contests, expected discrepancy rates for purposes of determining initial sample sizes, etc.
4. determine initial sample sizes based on contest margins and expected discrepancy rates
5. generate lists of ballots to be audited in each county, beginning with a seed created by a public process, using ballot manifests
6. upload interpretations of markings on individual ballots from each county ("audit CVRs")
7. compare marking interpretations with corresponding cast vote records for individual ballots, identify and track discrepancies
8. determine whether additional auditing is needed and specify additional ballots to be inspected
9. provide a web interface allowing the public to track the status of the audit overall and in detail
10. issue reports to the public and each county summarizing audit results

A crucial design goal is transparency: to enable the public to "audit the audit" without needing special software. (More on these points below.)

Transparency requirements

The main goal of risk-limiting audits is to support "evidence-based elections," which require "convincing affirmative evidence that the reported outcomes actually reflect how people voted."[1] To achieve this goal, every aspect of the audit must reflect design principles of being observable and verifiable by interested parties and the public. Based on those principles, we recommend the following transparency requirements for audit support software:

- All audit support software must be either open source (approved by the Open Source Initiative: see https://opensource.org/licenses), or at least "disclosed source" (in which vendors publicly disclose source code, but can still retain exclusive licensing rights).[2]

[1] "Evidence based elections," P.B. Stark and D.A. Wagner, 2012, http://www.stat.berkeley.edu/~stark/Preprints/evidenceVote12.pdf

[2] *Disclosure of Software for Voting Systems*, submission to *1st Symposium on Building Trust and Confidence in Voting Systems*,Dec 10-11, 2003, NIST, by Neal McBurnett, with updates. http://bcn.boulder.co.us/~neal/elections/disclosure.html

- The software should be cross-platform and run on a variety of common operating systems and web browsers, including Internet Explorer, Firefox, Chrome, and Safari.
- Accessibility: the software must comply with Section 508 requirements.
- The pseudo-random number generator must use the "sampler" algorithm, in order to interoperate with existing RLVTA software. Sampler is documented, with test cases, at https://github.com/cjerdonek/rivest-sampler-tests. The software must accept a seed value of up to 40 digits, generated in a public ceremony at the beginning of the audit, to use with sampler, and the seed should be included in the audit report so the sequence of pseudorandom selections can be independently verified.
- For sample size calculations, for both ballot comparison and ballot polling audits, the software should implement calculations described in "A Gentle Introduction to Risk-limiting Audits" by Mark Lindeman and Philip B. Stark (https://www.stat.berkeley.edu/~stark/Preprints/gentle12.pdf) and sources cited therein. Refer to https://www.stat.berkeley.edu/~stark/Java/Html/auditTools.htm and https://www.stat.berkeley.edu/~stark/Java/Html/ballotPollTools.htm for public implementations of the calculations.
- The software should be capable of easily incorporating new sampling methods (e.g., supporting stratified samples for CVR and non-CVR counties) in a manner that facilitates checking the implementation against technical descriptions similar to those cited in the previous point.
- The software must produce a detailed report that documents the audit evidence (see discussion under "Required functionality").
- The Audit Database (further described below) must be fully available for *read-only* access by the interested public.

Security requirements

The system must conform with the Colorado Information Security Policies (http://www.oit.state.co.us/ois/policies). Given the importance of the audit in documenting election results, it is crucial to implement rigorous security measures that restrict *write* access to the Audit Database to authorized users: presumably, county clerks and SoS designees. All changes to the Database shall be logged by time and user ID.

Required functionality

The list below is a partial enumeration of required functions:

- Support an Audit Database that consolidates ballot manifest information, Cast Vote Records (CVRs) from voting systems, and "audit CVRs" (interpretations by county election audit board judges of contest-specific vote marks on individual sampled ballots) uploaded from individual counties, and all other audit information from which the audit results are derived
- Combine county-level election reporting results from ENR system for all contests, ensuring that contest and choice names are consistent across counties (e.g., "Yes/For" vs. "Yes")
- Cast Vote Record (CVR) management
    - Receive CVR uploads from counties
    - CVR processing: re-tally and compare with ENR reports
    - Receive and publish audit CVRs from counties (possibly with associated free-form notes), and identify discrepancies with voting system CVRs
    - Support public download of voting system and audit CVRs, plus documentation of how discrepancies were investigated (see "Web interface…" below)

- Ballot manifest management
    - Support ballot manifest upload and download
    - Compare ballot manifests with CVRs and ENR data, report discrepancies
- RLA management
    - Support specification of risk limits by type of contest, and configuration of expected discrepancy rates for calculating initial sample sizes
    - Calculate initial sample size estimates for each county based on margins in each contest, risk limits and expected discrepancy rates, and the county distribution of ballots for each contest
    - Specify and publish lists of ballots to audit in each county, based on a publicly generated seed used with a PRNG (see Transparency Requirements), referencing the ballot manifests; indicate the contests to be audited on each ballot, including opportunistic auditing of contests with no mandatory risk limit
    - Support imprinted ballot identifiers as documented in CVRs or supplementary files
- Web interface for County Audit Board users
    - Provide an interface for entering audit CVRs (audit interpretations of the marks on each ballot in the contests to be audited), ideally presenting contests in the same order and with a similar layout as the original ballots, and allow for free-form notes
    - Guide users through a standard process to investigate discrepancies, documenting what steps were taken and any relevant conclusions
    - Provide straightforward navigation from screen to screen
    - Support upload of color photographs of the audited ballots (both/all sides, if applicable) via a smartphone app that associates photographs with the corresponding CVRs
    - Support simultaneous use by multiple counties
    - As a backup in case the interactive entry process is unavailable, support use of CSV files for uploading sets of audit CVRs
- Manage audit process
    - Track progress for each contest based on audit CVRs received so far; estimate remaining sample size (per contest by county, and for each county overall)
    - Determine when each contest subject to a risk limit has achieved that limit
    - Provide a web-based status “scoreboard” that allows the public to track progress per contest, per county, and overall statewide
    - Accept and track proposals for targeted audit units (ballots), and decisions on which ones to accept; manage the audit CVRs and results from targeted units separately from the randomly selected ballots
- Reporting and documentation
    - Produce a detailed audit report that documents which contests and ballots were audited, results including risk levels achieved for each contest, types of discrepancies (using predetermined codes) and how discrepancies were resolved, and any evidence needed to check the selections and risk calculations, with reference to the Audit Database
    - Produce metrics to help plan for and improve future audits