Non(c)esuch Ballot-Level Comparison Audits

1st EIS Workshop @ ESORICS
Copenhagen, DK

Philip B. Stark

30 September 2022

University of California, Berkeley

RLA: any procedure with a known minimum chance of correcting the reported election outcome if it's wrong, and never changes a correct outcome.

Risk limit: largest chance that the audit fails to correct a wrong outcome

Outcome: political outcome—who or what won—not the exact vote tally

Need trustworthy vote record: HMPB kept demonstrably secure throughout the election, canvass, and audit.

Need eligibility audits and compliance audits.

RLA sampling: individual cards and clusters of cards, with or without replacement, with or without stratification, with or without sampling weights, & Bernoulli sampling.

Many ways to use data from audited cards: polling audits, comparison audits, hybrid audits

Most efficient: *ballot-level comparison* (BLCRLA): compare how the voting equipment interpreted individual, randomly selected ballot cards (*cast-vote records*, CVRs) to how humans interpret the same cards (*manual-vote records*, MVRs).

BLCRLA requirement:

voting system commits to CVRs before the audit starts, in such a way that individual CVRs can be matched to individual ballot cards

Strategies for the matching:

- imprint IDs on cards after the cards have been disassociated from voters
- keep the cards in scan order (may require counting into stacks of 100s or 1000s of cards)

OK for CCOS and vote centers.

For PCOS, may compromise the anonymity of votes.

Workarounds:

- ballot-polling for the whole contest
  - much less efficient
- re-scan PCOS ballots centrally and base the audit on rescan
  - duplication of effort
  - checks the second tabulation, not the official tabulation
  - "lazy" audits (Harrison et al. 2022) only rescan batches from which ballots are selected for audit
- hybrid audits (Ottoboni et al. 2018) that use stratified sampling, w/ ballot-level comparison for CCOS and ballot polling for PCOS.
  - inefficient if there are many PCOS ballots
- ballot-level comparison for CCOS and batch-level comparison for PCOS, i.e., variable batch sizes from 1 to many
  - not as efficient as ballot-level comparison

Nonces

- Cryptographic *nonce*: number guaranteed to be unique (in some universe of numbers).
- Typically generated randomly or pseudo-randomly.
- Could serve as card IDs; randomness breaks link to voting order.

What if the imprinter . . .

- prints the same ID on more than one card?
- misreports the IDs it printed?
- fails to imprint an ID on one or more cards?

Plurality contest, Alice v. Bob.

Adversary wants Alice to win.

3 ballot cards were cast, 2 for Alice and 1 for Bob: Alice won.

Plurality contest, Alice v. Bob.

Adversary wants Alice to win.

3 ballot cards were cast, 2 for Alice and 1 for Bob: Alice won.

Adversary creates 3 CVRs: CVR w ID 17 has a vote for Alice; CVRs w IDs 91 and 202 have a vote for Bob.

According to CVRs, Bob won.

Plurality contest, Alice v. Bob.

Adversary wants Alice to win.

3 ballot cards were cast, 2 for Alice and 1 for Bob: Alice won.

Adversary creates 3 CVRs: CVR w ID 17 has a vote for Alice; CVRs w IDs 91 and 202 have a vote for Bob.

According to CVRs, Bob won.

Adversary prints the number 17 on the two cards with votes for Alice and the number 91 on the card with a vote for Bob.

Audit selects card at random, looks up the corresponding CVR.

Plurality contest, Alice v. Bob.

Adversary wants Alice to win.

3 ballot cards were cast, 2 for Alice and 1 for Bob: Alice won.

Adversary creates 3 CVRs: CVR w ID 17 has a vote for Alice; CVRs w IDs 91 and 202 have a vote for Bob.

According to CVRs, Bob won.

Adversary prints the number 17 on the two cards with votes for Alice and the number 91 on the card with a vote for Bob.

Audit selects card at random, looks up the corresponding CVR.

CVR will match the human reading of the card perfectly.

Thus, can't rely on sampling the cards (rather than the CVRs) w/o verifying that printed IDs are unique—a lot of work.

Sampling IDs and retrieving cards might work, but looking through a large pile of paper for a card imprinted with a particular nonce is hard.

Can we use tech w/o having to trust the tech?

- What if it returns a card with a different ID, or no card at all?
- What if more than one card is labeled with the same ID, and the tech picks the card to return adversarially?

Assumptions

- paper trail consists of every validly cast card; cards reflect the voters' actual selections.
- trustworthy upper bound on the total number of validly cast cards that contain the audited contests (e.g., from registration records, pollbooks, & physical ballot accounting)
- voting system, imprinting system, & card-retrieval system untrusted.

How might this overall system misbehave?

1. number of CVRs that contain the contest might not equal number of cards that contain the contest
2. CVRs might misrepresent votes on one or more cards
3. imprinting might omit or repeat IDs in the CVR list, print IDs that are not in the CVR list, or fail to imprint any ID on some cards
4. retrieval system could return a card w different ID than the ID requested, a card with no ID, or no card.

Goal: an audit procedure that has a guaranteed minimum chance $1 - \alpha$ of becoming full hand count if the outcome according to the CVRs differs from the outcome according to the votes on the cards

SHANGRLA assorters

SHANGRLA (Stark, 2020) reduces RLAs to testing whether the averages of a collection of finite lists of nonnegative, bounded numbers are all greater than $1/2$.

Each list results from applying an "assorter" A to the votes in a contest for each validly cast card that contains the contest

Value assorter A assigns to card b_i is $A(b_i) \in [0, u]$; the value A assigns to the votes in CVR c_j is $A(c_j) \in [0, u]$.

- $\bar{A}^b := \frac{1}{N_b} \sum_i A(b_i)$
- $\bar{A}^c := \frac{1}{N_c} \sum_i A(c_i)$
- $v := 2\bar{A}^c - 1$, *reported assorter margin*

Suppose there is a 1:1 mapping between ballot cards and CVRs for a given contest.

overstatement assorter

$$B(b_i, c_i) := \frac{1}{2 - v/u} \left[1 - \frac{1}{u}(A(c_i) - A(b_i)) \right] \in [0, 2/(2 - v/u)].$$

Then $\bar{A}^b > 1/2$ iff $\bar{B} := \frac{1}{N_b} \sum_i B(b_i, c_i) > 1/2$.

Let π be any permutation of $\{1, \ldots, N_b\}$.

The mean of a list does not depend on its order, so

$$\bar{A}^b = \frac{1}{N_b} \sum_i A(b_i) = \frac{1}{N_b} \sum_i A(b_{\pi(i)})$$

and

$$\bar{A}^c = \frac{1}{N_b} \sum_i A(c_i) = \frac{1}{N_b} \sum_i A(c_{\pi(i)}).$$

Define

$$\bar{B}^\pi := \frac{1}{N_b} \sum_i B(b_{\pi(i)}, c_i).$$

Then

$$\bar{B^\pi} = \bar{B}.$$

Thus, $\bar{A}^b > 1/2$ iff $\bar{B^\pi} > 1/2$ for *any* π.

What if #cards $\neq$ #CVRs?

Suppose $N_b < N_c$.

Then there has been a malfunction, a procedural error, or the integrity of the paper trail was compromised.

If compliance audit confirms the paper trail is trustworthy, can still audit by ignoring the contest on *any* $N_c - N_b$ CVRs that contain the contest, *provided* $\bar{A}^c > 1/2$ after those CVRs are omitted.

Let π be *any* 1:1 mapping between the remaining CVRs & the cards (even adversarial)

Then $\bar{B}^\pi > 1/2$ implies $\bar{A}^b > 1/2$.

Now suppose $N_c < N_b$.

Auditors can create $N_b - N_c$ "phantom" CVRs w no valid vote in the contest.

phantoms won't have IDs; if the audit selects one, set $A(c_i) := 1/2$.

Let π be *any* 1:1 mapping between those CVRs and those ballot cards, even a mapping created by a malicious adversary.

Outcome is correct if

$$\bar{B^{\pi}} > 1/2$$

for every overstatement assorter.

Untrustworthy imprinting and retrieval

- assume wlog $N_b = N_c$ (or make it so, as described)
- construct a canonical 1:1 mapping π from CVRs to cards
- define a deterministic function that couples sampling from $\{B(b_{\pi(i)}, c_i)\}_{i=1}^{N_b}$ to sampling from a related population $\{L_i\}_{i=1}^{N_b}$ for which $0 \leq L_i \leq B(b_{\pi(i)}, c_i)$ for all i.
- since $L_i \leq B(b_{\pi(i)}, c_i)$, $\bar{L} := \frac{1}{N_b} \sum_i L_i \leq \bar{B}$
- hence if $\bar{L} > 1/2$, also $\bar{B} > 1/2$.
- test $\bar{L} \leq 1/2$, e.g. using methods in SHANGRLA or ALPHA (Stark, 2022)

The canonical mapping π.

- For each CVR ID ζ:
  - let i denote the index of the CVR with ID ζ
  - if ζ is on exactly one card, $\pi(i)$ is the index of that card.
  - if ζ is on more than one card, $\pi(i)$ is the index of the card in that set that maximizes $B(b_{\pi(i)}, c_i)$, w ties broken arbitrarily.
- If some ID does not appear on any card, there are leftover IDs and an equal number of leftover cards. Let π pairs their indices arbitrarily.

π is a 1:1 mapping from CVR indices to cards, so $\bar{A} > 1/2$ iff $\bar{B^{\pi}} > 1/2$.

Coupling

- if select a CVR w ID ζ for audit, system might not retrieve card w ID ζ, so can't calculate $B(b_{\pi(i)}, c_i)$.
- construct lower bound $L_i \leq B(b_{\pi(i)}, c_i)$ from CVR c_i with ID ζ and whatever card the system retrieves (or none at all)
- if the system returns a card b w ID ζ, then ζ was imprinted on at least one card.
  - if ζ imprinted on *exactly* one card, $B(b, c_i) = B(b_{\pi(i)}, c_i)$
  - if ζ was imprinted on more than one card, then $B(b, c_i) \leq B(b_{\pi(i)}, c_i)$, since π was constructed to maximize $B(b_{\pi(i)}, c_i)$ over all cards $\{b_j\}$ w ID ζ.
  - thus, if system returns a card w ID ζ, can take $L_i := B(b, c_i) \leq B(b_{\pi(i)}, c_i)$.
- if system does not return a card or returns card w no ID or any ID $\neq \zeta$, the true value of $B(b_{\pi(i)}, c_i) \geq \frac{1-A(c_i)/u}{2-v/u} =: L_i$.

If the imprinting and retrieval do what they are supposed to do, $L_i = B(b_{\pi(i)}, c_i)$ for all i: no workload penalty for the protection against misbehavior.

Full procedure

- The voting system commits to a set of CVRs, each with an ID ζ, and commits to an ID ζ (possibly blank) on each card by printing that ID on the card.
- Inputs: CVRs with IDs, risk limit, upper bound on #cards that contain each contest under audit, reported winners for each contest
- create assorters & overstatement assorters for every contest; select risk-measuring function
- check whether $\bar{A}^c > 1/2$ for all assorters. If not, audit fails.
- check whether $\{\zeta_i\}_{i=1}^{N_b}$ are unique. If not, the audit fails.
- check whether the number of CVRs that contain each contest equals the number of ballot cards that contain that contest. If not, alter CVRs or create phantom CVRs.
- pick seed for the audit's PRNG

- While at least one assertion is marked 'unconfirmed':
  - Select an ID ζ at random from the CVR list. Let i denote the index of the CVR c_i with that ID.
  - If a card with ID ζ has been requested before, use the card (if any) previously retrieved. Otherwise, request card with ID ζ.
  - If all cards have been retrieved, determine the correct outcome of every contest from the cards and terminate the audit. Otherwise:
    - For every contest on c_i under audit, for every A for that contest not yet been confirmed, calculate corresponding L_i
    - Update the measured risk for every as-yet unconfirmed assertion using the corresponding value of L_i.
  - mark all assertions for which the measured risk is less than or equal to the risk limit for the corresponding contest 'confirmed.'
  - mark as 'confirmed' all contests for which all assertions have been marked 'confirmed'
  - At any time, can do full hand count, for instance if it might be less work. Outcome according to the hand count becomes the final outcome; audit ends.
- End audit: all assertions have been confirmed.

Implementation considerations

- seed used to generate the nonces should include entropy that even an insider cannot know to the order in which ballot cards were cast.
- system should not timestamp the CVRs or digital images (or the files that contain them)
- crucial that the imprinter cannot create, alter, or obfuscate votes. For instance, the imprinter should only use red or green ink and should not be physically able to print near any vote target
- to facilitate the automated retrieval of cards with particular IDs, use OCR-friendly font. Barcodes or QR codes could be used, since it is not essential to the risk-limiting property that the identifiers be human-readable

Note: this method protects the risk limit, but a faulty implementation (e.g., using predictable identifiers instead of genuine nonces) could compromise privacy.