White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of York.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/245062/

Version: Accepted Version

Proceedings Paper:
Nolan, Jack, Vazirinejad, Yasmine, Shahandashti, Siamak F. et al. (Accepted: 2026) Peel and Vote: A Paper Ballot Design for Publicly Verifiable Risk-Limiting Audits. In: Eleventh International Joint Conference on Electronic Voting (E-Vote-ID 2026). Eleventh International Joint Conference on Electronic Voting, 06-09 Oct 2026 , EST. (In Press)

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

Peel and Vote: A Paper Ballot Design for Publicly Verifiable Risk-Limiting Audits

Jack Nolan[1], Yasmine Vazirinejad[1], Siamak Shahandashti[2], and Feng Hao [1]

Abstract:

Post-election Risk-Limiting Audits (RLAs) provide statistical guarantees of election outcomes by hand-counting randomly selected paper ballots. However, standard RLAs assume that ballots remain unaltered in storage between election day and the audit, but voters cannot verify this assumption. Prior frameworks, such as VAULT (E-VOTE-ID 2019), attempt to make the audit process verifiable, but they require the public to trust the Election Authority (EA) to honestly sample ballots at random and correctly interpret the voters' choices into cryptographic commitments. Furthermore, they are typically *ballot-level*, requiring a 1-to-1 mapping between individual paper ballots and digital records. The EA also needs to interactively open commitments during the audit. To address these limitations, we propose the first verifiable *batch-level* RLA scheme, which shifts the trust anchor from the EA directly to the voter and operates on groups of ballots (batches) to eliminate individual tracking hurdles. Our scheme introduces a peel-and-vote paper ballot design with removable flaps containing pre-printed cryptograms. A voter selects a candidate by removing the corresponding flap, and retains the flap as a receipt, while the same cryptogram printed on the receipt is also published on a bulletin board. In our scheme, a *batch* contains a group of cryptographically related ballots that are randomly distributed across the precincts before the election. Once all cryptograms in a batch are published, anyone can independently compute the tally for that batch, achieving a *self-tallying* property. Intuitively, our scheme transforms a national-scale election into many smaller self-tallying elections: one for each batch. Based on historical US election data, we show that when the ballot loss rates remain below 0.5%, the scheme enables tally verification for 61% of randomly selected ballots with a batch size of 100, or 37% with a batch size of 200. These figures are substantially higher than the 0.01–0.3% sampling coverage typically achieved in RLAs. This improvement stems from the fact that auditing is built directly into the paper ballot design in our scheme as an always-on feature, rather than being treated as an optional manual post-election process.

Keywords: Verifiability, Risk-limiting auditing, Paper ballots

1 Introduction

The integrity of a democratic election relies on the public's trust in its outcome. To achieve scalable tallying, many countries, such as the United States, use a hybrid system in elections: paper ballots capture voter intent, while optical scanners digitise ballots into Cast Vote Records (CVRs) to electronically count the votes. However, electronic tallies cannot be blindly trusted: software bugs, miscalibrated sensors, and malicious tampering can compromise the machine count [Wa21]. To mitigate these risks, jurisdictions employ

[1] University, of, Warwick, jack.nolan@warwick.ac.uk; yasmine.vazirinejad@warwick.ac.uk; feng.hao@warwick.ac.uk, https://orcid.org/0000-0002-8664-5074
[2] University, of, York, siamak.shahandashti@york.ac.uk

post-election audits. The gold standard is the *Risk-Limiting Audit* (RLA), first applied to elections by Stark [St08] and further formalized in subsequent works [Go; LS12]. An RLA is a sequential statistical method [Wa92] that hand-counts a sample of paper ballots or other Voter-Verifiable Paper Records (VVPRs) to mathematically guarantee that the probability of certifying an incorrect outcome is bounded by a prescribed *risk limit* $\alpha \in [0, 1]$. In other words, given a reported election outcome and risk limit α, an RLA guarantees that the probability of failing to correct an incorrect outcome is at most α.

An RLA can be formulated as a set of hypothesis tests at significance level equal to the risk limit [St20]. For each hypothesis test, a test statistic is initialised and its critical region is calculated. Ballots are then sequentially sampled and manually interpreted as a particular vote (or as an invalid vote). Each hypothesis test is followed to update the test statistics according to the ballot's interpretation. If a test statistic enters its critical region, the corresponding null hypothesis is rejected. If every null hypothesis is rejected, the RLA has concluded and has asserted the election outcome at the risk-limit. If the number of sampled ballots exceeds some predefined maximum, then a full hand-count is ordered. In this way, the RLA can be viewed as an algorithm that samples ballots until either every null hypothesis is rejected (accept the reported election outcome), or a certain number of ballots have been sampled (perform a full hand-count).

In a *polling audit* [LS12], sampled ballots alone are used to update the test statistics. In a *comparison audit*, sampled ballots are directly compared to the CVRs. In a *batch-level audit*, vote tallies from groups of ballots (called batches) are sampled and used to update test statistics, rather than individual ballots as in a *ballot-level audit*. Throughout this work, we focus on batch-level polling audits, such as BRAVO [LSY12].

The Open Problem: The Physical Trust Gap. While RLAs provide a statistical framework for error detection, they rely on the physical chain-of-custody of the ballot boxes. If the public suspects that paper ballots have been added, removed, or altered in storage between election day and the audit, the mathematics of the RLA becomes meaningless. An audit cannot assert the election outcome if the physical paper trail itself is not anchored to a cryptographic truth established at the moment of ballot casting.

Prior works have attempted to close this transparency gap by combining RLAs with cryptography to achieve verifiability. The SOBA framework [Be11] pioneered the use of hash commitments to allow public verification of audits but suffers from poor-usability due to its complexity. Later, VAULT [BST19] utilised homomorphic encryption to secure digital commitments for comparison audits. While VAULT elegantly handles privacy, it suffers from two critical limitations. First, it relies on a fundamental interpretation of trust: the public must trust that the Election Authority (EA) honestly interpreted and digitized the physical paper markings into the cryptographic commitments, but a voter has no way to ensure that their intent was not flipped before the commitment phase. Second, VAULT is inherently interactive: during the audit, the EA must actively publish the randomness to

open specific commitments. If the EA is uncooperative or its systems are compromised post-election, the audit stalls and is forced to default to the worst-case statistical penalties.

Our Contributions. To address these limitations, we propose the first non-interactive, batch-level voter-verifiable RLA scheme for plurality contests. By combining a physical flap-based ballot design with DRE-i style pre-computed cryptograms [Ha14] (printed cryptographic information, which we represent as QR codes), we remove the need for blind trust in the electronic tallying and auditing process. Thus, the interpretation and logistical hurdles are solved simultaneously. Specifically:

- We introduce a paper ballot design featuring removable flaps. By removing exactly one flap to cast a vote, voters retain a receipt and can probabilistically verify the integrity of the pre-printed cryptograms using a Benaloh Challenge [Be06] prior to casting. A cryptogram includes an encrypted vote for the choice of the candidate and a 1-out-of-C Non-Interactive Zero-Knowledge (NIZK) proof of well-formedness [CDS94].
- We construct the ballot cryptography such that the random blinding factors within any batch naturally sum to zero. By publishing these encrypted votes to a public Bulletin Board, the system achieves universal verifiability: the public can aggregate the encrypted votes to deduce a single encoded batch tally, from which individual candidate tallies can be perfectly extracted. Crucially, this is achieved without requiring the EA to actively decrypt or open any commitments.
- We evaluate the practical feasibility of our scheme using historical data from United States elections, demonstrating that our verifiability mechanisms and the *BRAVO* RLA algorithm [LSY12] can be successfully executed under real-world conditions.

2 Scheme Overview

We consider a plurality election with multiple candidates, from which voters choose only one. The presented voting method requires paper ballots to be equipped with removable *flaps*: one per candidate. Removing exactly one flap indicates a voter's choice for a candidate, with the flap serving as the voter's receipt which they may keep. Cast ballots are scanned, and cryptographic information from each scanned ballot is published to a *bulletin board* (a publicly accessible, append-only electronic ledger) that voters can check against their receipt. The design allows anyone with access to the bulletin board to conduct a batch-level Risk-Limiting Audit (RLA).

2.1 Ballot Construction

Before ballot production, an upper bound for the number of voters n must be known. The Election Authority (EA) prints $N = \kappa n$ ballots ($\kappa > 1$ allows extra ballots to be used for

auditing) which are identified by a serial number $i \in [N]$ printed on the ballot (we denote $[m] = \{1, \ldots m\}$ and $[m]_0 = \{0, \ldots, m\}$). Voters are assigned to ballots via an injection $\rho : [n] \to [N]$. To prevent anyone from linking voters to their ballot serial numbers, ρ is instantiated physically by randomly shuffling the printed ballots before distribution. This assignment must remain uniformly random to prevent batch-manipulation attacks (§5.1).

Ballots are randomly partitioned into K batches of sizes $(N_k)_{k=1}^{K}$ that sum to N. We represent this assignment as a map $A : [N] \to [K]$ so that ballot i is in batch $A(i)$. We also define $a(i) = A^{-1}(A(i))$ as the set of all ballots in the same batch as ballot i. Batches are chosen to have approximately equal size, so that $N_k \approx N/K$.

For each of the C candidates, a pre-computed cryptogram [Ha14] is printed next to the candidate's name on the ballot body, as well as on the hidden side of the attached flap. The cryptogram contains an encrypted vote with a zero-knowledge proof to show well-formedness of the ciphertext. Ballots must be designed to be *tamper-proof*: it should be impossible to fully read any cryptogram without completely removing a flap, and removing a flap even partially must produce evidence of tampering. Figure 1 shows an example ballot.

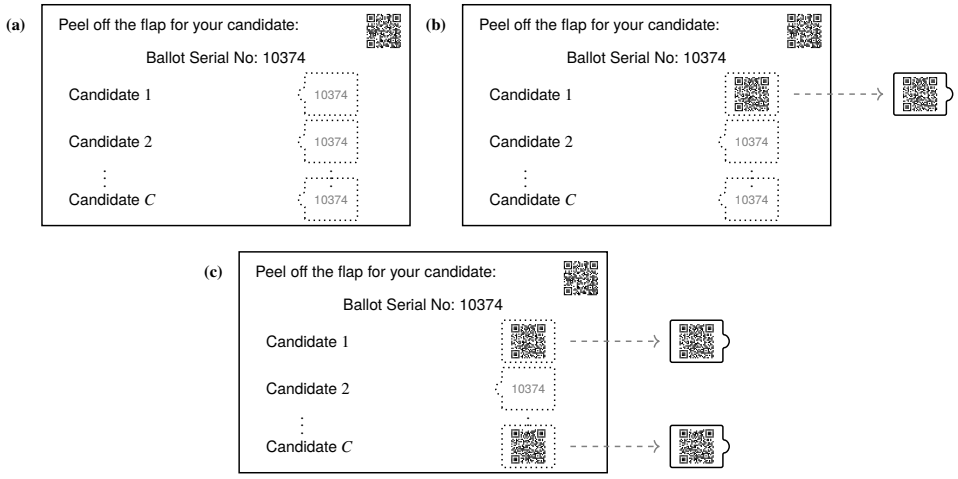


Fig. 1: Simplified Ballot Construction. (a) with all flaps intact, (b) after one flap is peeled off, representing a valid cast ballot and (c) represents an invalid ballot after more than one flap is removed. Each flap has the ballot serial number i printed on its visible side and a cryptogram printed on the other side as a QR code encoding the encrypted vote $V_{i,c}$ and a zero-knowledge proof $\pi_{i,c}$. The QR code in the top-right corner contains commitments (g^{x_i}, g^{y_i}) and a digital signature.

The cryptography operates in a subgroup $\mathcal{G}_q \subset \mathbb{Z}_p^*$ of prime order q generated by g. For each ballot $i \in [N]$, the EA uniformly randomly samples a private key $x_i \in \mathbb{Z}_q \backslash \{0\}$ and computes $y_i = \sum_{j \in a(i),\ j>i} x_j - \sum_{j \in a(i),\ j<i} x_j \pmod q$. Note that y_i is calculated by a summation strictly over voters in the same batch as i, as opposed to every voter as in DRE-i.

This localized summation ensures the following equality holds for all batches $k \in [K]$, which enables our non-interactive tallying:

$$\sum_{i:A(i)=k} x_i y_i = 0 \pmod{q}. \tag{1}$$

For each candidate $c \in [C]$ on ballot i, the EA computes the encrypted vote $V_{i,c} \equiv g^{x_i y_i} g^{E(c)} \pmod{p}$, where $E : [C] \to \mathbb{Z}_q$ is an injection used for *vote-encoding* to allow individual tallies to be *uniquely* deduced from a single aggregate. For E to be a valid encoding, there must exist a decoding map $\mathcal{T} : \mathbb{Z}_q \to [N]_0^C$ such that for all batches $k \in [K]$ and possible tallies $T_{k,c} \in [N_k]_0$ subject to $\sum_{c\in[C]} T_{k,c} \leq N_k$, we have $\mathcal{T}(\sum_{c\in[C]} T_{k,c} \cdot E(c)) = (T_{k,c})_{c\in[C]}$. This can be efficiently achieved by:

$$E(c) = (N^* + 1)^{c-1}, \tag{2}$$

where $N^* = \max_k\{N_k\}$ is the largest batch size. In Eq. 2, $E(c)$ is defined such that $\sum_{c\in[C]} T_{k,c} \cdot E(c)$ represents a base-$(N^* + 1)$ integer with C unique digits $(T_{k,c})_{c\in[C]}$. Other choices of $E(c)$ are possible [Cr96].

Each encrypted vote is supplemented by a *proof of well-formedness* $\pi_{i,c}$: The EA commits to x_i, y_i by printing $X_i = g^{x_i}$ and $Y_i = g^{y_i}$ on the ballot, and forms a proof $\pi_{i,c}$ as a 1-out-of-C Non-Interactive Zero-Knowledge (NIZK) proof [CDS94] that $V_{i,c}$ satisfies the predicate:

$$P(V_{i,c}) \iff \bigvee_{j\in[C]} \left(\log_g(Y_i) = \log_{X_i}\left(V_{i,c}/g^{E(j)}\right)\right).$$

The pair $(V_{i,c}, \pi_{i,c})$ forms the complete cryptogram printed on flap c on ballot i. The construction of $\pi_{i,c}$ using the Fiat-Shamir heuristic [FS86] is detailed in §6.

Finally, the ballot's visible contents are signed by the EA to create a ballot signature σ_i, which is printed on the ballot. This can be used to confirm the authenticity of the ballot using the EA's public key $\mathrm{PK}_{\mathrm{EA}}$, preventing ballot-stuffing attacks.

2.2 Voting

Ballot Auditing. Before voting starts, any party observer in the polling station can audit paper ballots under supervision as follows. They randomly choose a set of paper ballots. For each ballot, they remove **all** flaps to reveal the QR codes containing the cryptograms. They check that for each flap, the QR code printed on the back of the flap matches that printed on the paper ballot from where the flap was removed (see Figure 1). Based on the revealed cryptograms, anyone can verify whether the encrypted vote represents a valid choice of candidates. As an example, assume there are $C = 3$ candidates in the election with the encoding $E(c)$ for each candidate as defined in Eq. 2. The cryptograms on an audited

ballot i for the three candidates $c = 1, 2, 3$ contain the encrypted votes as follows (omitting mod p):

$$V_{i,1} = g^{x_i y_i} g, \tag{3}$$

$$V_{i,2} = g^{x_i y_i} g^{N^*+1}, \tag{4}$$

$$V_{i,3} = g^{x_i y_i} g^{(N^*+1)^2} \tag{5}$$

The associated ZKP for each cryptogram proves that the encrypted value is one of $\{E(1), E(2), E(3)\}$ (i.e., $\{1, N^*+1, (N^*+1)^2\}$) without revealing which. However, once all three cryptograms are revealed, it becomes trivial to decrypt the votes based on their rational relationships. That is, the encrypted value for the second candidate is the encrypted value for the first candidate multiplied by g^{N^*}, and the encrypted value for the third candidate is the encrypted value for the second candidate multiplied by $g^{(N^*+1)^2-N^*-1}$. Party observers can repeat the above process to audit any randomly chosen ballot and ensure that the cryptograms are valid. This establishes public assurance that the paper ballots are well-formed. All audited ballots will be placed in the **audit box**. Note that auditing the paper ballot *does not* require any secret key. The verification of the ZKPs and the correct correspondence of the cryptograms to the candidates can be done publicly without any secret keys.

The ballot auditing establishes the assurance that the cryptograms correspond to the correct candidates. Once the correspondence between the cryptograms and the candidates is established, it is easy to compute $g^{x_i y_i}$ for the ballot i. For an audited ballot i, only $g^{x_i y_i}$ is included in the tallying process. The value of $g^{x_i y_i}$ effectively adds 0 to the tally, hence not affecting the election result at all. If an observer removes some flaps (> 1), instead of all, the auditing works basically the same. Based on the revealed cryptograms, their correspondence to the candidates can be established based on the rational relationships among the encrypted values. Subsequently, $g^{x_i y_i}$ for the audited ballot i is computed and then included in the tallying process.

Ballot Casting. On election day, authenticated voters complete their ballots in private. To cast a vote for candidate c, the voter removes the corresponding flap. The voter places the ballot in the **cast box**, retaining the removed flap as a receipt containing i, $V_{i,c}$ and $\pi_{i,c}$. They can later use this to verify that the same data as printed on the receipt is published on the public bulletin board as the recorded entry for ballot i.

A voter may also audit their paper ballot before voting starts. To do this, they should first inform the election staff in the polling station of their intent to do auditing and then audit the ballot publicly by removing all flaps. The voter does not need to cryptographically verify the audited ballot by themselves. They retain the removed flaps as receipts; they just need to ensure the same data on the receipts is published on the bulletin board. Anyone with read access to the bulletin board will be able to perform universal verification on the revealed cryptograms for the audited ballots. Any audited ballots are put into the **audit box**. The voter is then issued a new ballot as in the Benaloh Challenge [Be06].

2.3 Scanning and Tallying

Blank ballots at the end of the voting deadline have all flaps removed under the supervision of party observers and are treated as audited. Cast and audited ballot boxes are securely transported to a count centre and unsealed under supervision from auditors and interested parties. A cast ballot is valid if exactly one flap has been removed. Only the valid ballots contribute to the tally of the candidates, and all other ballots are invalid for counting.

Invalid Cast Ballots: Overvotes (multiple flaps removed) are transferred to the Audit Box unaltered. Undervotes (no flaps removed) have all flaps manually removed by the EA under supervision before also being placed in the audit box.

The two boxes partition the set of all ballots $[N]$ into cast ($\mathbb{C}$) and audited/invalid ($\mathbb{A}$) ballots, all of which can now be scanned so that votes can be tallied. The scanner initialises the tally vector $\hat{\boldsymbol{T}} = 0^C$. For each ballot, the scanner captures the identifier i, batch k, and commitments $\hat{\boldsymbol{X}}_i = (\hat{X}_i, \hat{Y}_i)$. To show that ballots might not be scanned correctly, we put a circumflex ($\hat{}$) over variables that have been obtained by scanning. For example, $\hat{X}_i$ represents the scanner's interpretation of the printed value X_i.

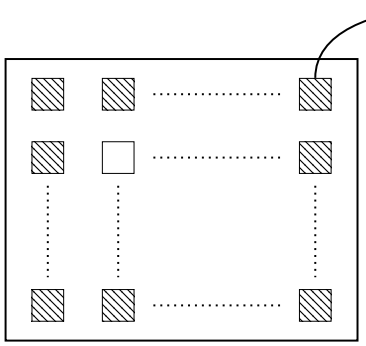
For a cast ballot $B_i \in \mathbb{C}$, the scanner reads the physical flaps to determine the plaintext choice c_i. It then increments $\hat{T}_{c_i}$ and securely deletes c_i from memory to preserve privacy. It then captures the single encrypted vote $\hat{V}_i$ and proof $\hat{\pi}_i$ corresponding to the removed flap. For an audited or invalid ballot $B_i \in \mathbb{A}$, the scanner captures the full tuple of all revealed cryptograms in order $\hat{\boldsymbol{V}}_i = (\hat{V}_{i,1}, \ldots, \hat{V}_{i,C})$ and proofs $\hat{\boldsymbol{\pi}}_i$. If an invalid ballot has fewer than C flaps removed, the unrevealed cryptograms and proofs are set to $\perp$. The audited and invalid ballots do not contribute to the tally.

Bulletin Board Entries. To support universal verifiability, every scanned ballot generates a bulletin board entry, with the field $status \in \{\text{Cast}, \text{Audit}, \text{Invalid}\}$ used to describe the ballot. The bulletin board entry for ballot i is a tuple:

$$(i, k, [\hat{V}_i, \hat{\pi}_i], \hat{\boldsymbol{X}}_i, status)$$

1. **Cast ballots** ($status$ = Cast): $[\hat{V}_i, \hat{\pi}_i] = \hat{V}_i, \hat{\pi}_i$.
2. **Audited/Invalid ballots** ($status$ = Audit/Invalid): $[\hat{V}_i, \hat{\pi}_i] = \hat{\boldsymbol{V}}_i, \hat{\boldsymbol{\pi}}_i$.

In edge cases where physical ballots are damaged and cannot be scanned, they must be manually adjudicated. Consequently, these ballots lose their public verifiability, emphasising the need for high-quality printing and scanning hardware. When scanning is complete, the full tally matrix $\hat{\boldsymbol{T}}$ is published. The announced winner is the candidate with the largest number of votes ($\arg\max_c(\hat{T}_c)$). Figure 2 illustrates the complete bulletin board, which is signed by the EA to confirm its authenticity.



(a) Full set of ballots: shaded boxes represent published ballots and unshaded boxes represent unpublished (missing) ballots; Each row represents a batch.

Election parameters: $p, q, g, N, A, \mathrm{PK}_{\mathrm{EA}}$

Serial	Encrypted vote	Proof	status
101	$\hat{V}_{101}$	$\hat{\pi}_{101}$	Cast
⋮	⋮	⋮	⋮
149	$\hat{\boldsymbol{V}}_{149}$	$\hat{\boldsymbol{\pi}}_{149}$	Invalid
⋮	⋮	⋮	⋮
160	$\hat{\boldsymbol{V}}_{160}$	$\hat{\boldsymbol{\pi}}_{160}$	Audited
⋮	⋮	⋮	⋮
200	⋮	⋮	⋮

Candidate	1	⋯	C
Tally	$\hat{T}_1$	⋯	$\hat{T}_C$

(b) The bulletin board for a complete batch and the final tallies

Fig. 2: The bulletin board: Each complete batch (full row of shaded boxes in (a)) has its ballots published and the final vote tallies are reported as in (b).

3 Verifiability

Verifiability ensures both individuals and the public can audit the election's integrity. It is broken down into *individual* and *universal* verifiability.

Individual Verifiability. The ability for voters to audit their ballot(s) as many times as desired through the Benaloh Challenge provides the cast-as-intended verifiability guarantee.

After casting their vote, the voter can match their retained physical flap with an entry on the public bulletin board by querying the bulletin board for ballot ID i, confirming its status as "Cast", and ensuring that the printed cryptogram on their flap perfectly matches the part of the bulletin board entry $(\hat{V}_i, \hat{\pi}_i)$. This procedure provides the recorded-as-cast verifiability guarantee. Because the board only records encrypted votes, the voter cannot prove their plaintext vote to a third party, preserving receipt-freeness.

Universal Verifiability. Anyone from the public can validate bulletin board entries. For every published bulletin board entry $(i, k, [\hat{V}_i, \hat{\pi}_i], \hat{\boldsymbol{X}}, \mathit{status})$, observers verify three conditions: (1) the ballot index $i \in [N]$ is unique, and $A(i) = k$; (2) the bulletin board's digital signature is valid; and (3) the NIZK proofs of well-formedness (§ 6) verify correctly. For status = Cast ballots, the single proof $\hat{\pi}_i$ is verified against $\hat{V}_i$. For status = Audit/Invalid ballots, all published proofs $\hat{\boldsymbol{\pi}}_i$ are verified against $\hat{\boldsymbol{V}}_i$.

The published encrypted vote tuple $\hat{\boldsymbol{V}}_i$ for a $status$ = Audit/Invalid ballot i can also have its permutation verified: $\hat{\boldsymbol{V}}_i$ is divided element-wise by the tuple $(g^{E(c)})_{c\in[C]}$, ignoring any elements that are $\perp$. The permutation is verified if the resulting tuple $\boldsymbol{U}_i := ((\hat{\boldsymbol{V}}_i)_c / g^{E(c)})_{c\in[C]}$ contains a single repeated value U_i. Widespread failures indicate a compromised election. This procedure provides the well-formedness verifiability guarantee for all recorded ballots.

4 Auditing

Our scheme allows a member of the public to follow *any* batch-level polling RLA algorithm $\mathcal{R}$. For random batch sampling, the sampling frame is established from the bulletin board: If a batch k is such that *every* ballot $i \in A^{-1}(k)$ has an entry on the bulletin board, we say batch k is *complete* and include it in the sampling frame. Also determined by $\mathcal{R}$ is the maximum number of batches $K_{\max} < K$ that may be sampled before a full hand-count is ordered. The audit is initialised as usual, by determining the null hypotheses and initialising the test statistics. Batches are then sampled from the sampling frame until either every null hypothesis is rejected, or $K_{\max}$ batches have been sampled. Since ballots are assigned to batches randomly, complete batches are randomly distributed, justifying our restriction of the sampling frame to complete batches only.

When a batch k is sampled, its vote tallies must be determined to update the test statistics. Instead of manual interpretation of VVPRs as in a traditional RLA, our system determines voter-verifiable batch tallies using the public bulletin board. Letting $T_{k,c}$ denote the number of votes for candidate c from batch k, the tallies from batch k are determined as follows as follows:

1. For each ballot $i \in A^{-1}(k)$ on the bulletin board, derive a *tally contribution* U_i: for $status$ = Cast ballots, U_i is simply the encrypted vote $\hat{V}_i$; for $status$ = Audit/Invalid ballots, U_i is obtained by dividing any published encrypted vote by g to the power of the corresponding candidate encoding $\hat{V}_{i,c} / g^{E(c)}$, as in the permutation verification.
2. Compute the *Batch Product* $P_k = \prod_{i\in A^{-1}(k)} U_i$.
3. Extract the *Aggregate Tally Value* $S_k = \log_g(P_k)$.
4. Decode the tallies using the decoding map: $\mathcal{T}(S_k) = (T_{k,c})_{c\in[C]}$.

Explanation of Correctness. If all ballots from batch k are scanned correctly, then $\hat{\boldsymbol{V}}_i = \boldsymbol{V}_i$ for all $i \in A^{-1}(k)$. Then for for $status$ = Cast ballots, $U_i = g^{x_i y_i} g^{E(c)}$, and for $status$ = Audit/Invalid ballots $U_i = g^{x_i y_i}$.

For a $status$ = Audit/Invalid ballot i, the permutation verification will detect if a nontrivial permutation σ is used to reorder encrypted vote tuples. The verification succeeds if the tuple $\boldsymbol{U}_i$ contains a single repeated value. That is, for all distinct $c, c' \in [C]$, we have $\hat{V}_{i,\sigma(c)} / g^{E(c)} = \hat{V}_{i,\sigma(c')} / g^{E(c')}$. If the cryptograms were scanned correctly, this yields

$E(\sigma(c)) - E(c) = E(\sigma(c')) - E(c')$. Since the vote encoding E must admit a map $\mathcal{T}$ that *uniquely* decodes vote aggregates, this is only possible if $\sigma(c) = c$.

Adopting the shorthand $\mathbb{C}_k = \mathbb{C} \cap A^{-1}(k)$ as the set of cast ballots from batch k, we have

$$P_k = \left(\prod_{i \in A^{-1}(k)} g^{x_i y_i} \right) \cdot \left(\prod_{i \in \mathbb{C}_k} g^{E(c_i)} \right) = g^{\left(\sum_{c \in [C]} T_{k,c} \cdot E(c) \right)},$$

since by construction (Eq. 1), the blinding factors $g^{x_i y_i}$ multiply to $g^0 = 1$.

Then $S_k = \sum_{c \in [C]} T_{k,c} \cdot E(c)$ and $\mathcal{T}(S_k) = (T_{k,c})_{c \in [C]}$. Because the total number of votes $\sum_{c \in [C]} T_{k,c} \leq N_k$ is bounded, S_k falls within a computationally feasible range to compute the discrete-logarithm, so Step 3 above can be computed efficiently in $O(\sqrt{N_k \max(E)})$ time (where $\max(E) = \max\{E(c) : c \in [C]\}$ is the largest possible value of E) using the *baby-step giant-step* algorithm [BT00].

The batch tallies $(T_{k,c})_{c \in [C]}$ for batch k are provided to $\mathcal{R}$ to progress the RLA until it terminates. On termination, either the reported election results will be asserted at a certain risk limit, or a full hand-count will be ordered.

5 Feasibility Analysis

If any ballot i does not create an entry on the bulletin board, its batch $k = A(i)$ cannot be added to the RLA sample frame. If the RLA algorithm exhausts the sample frame without asserting the election outcome, a full hand count will be required. Assuming that each ballot is "lost" with some fixed likelihood, the probability that batch k is complete is maximised by minimising the batch size N_k.

If every voter in batch k votes uniformly (all for the same candidate), then a voter in that batch can use their receipt to prove their vote to a third party, violating the *receipt-freeness* privacy property [DKR06]. The probability of this violation is minimised by maximising N_k.

In this section, we make these notions rigorous and show that both desirable properties can be achieved with high probability using historical data from US presidential and house elections.

5.1 Confidentiality

Uniformity-Freeness. It may happen that every voter from a complete batch k votes for the same candidate c. In this situation, each voter can use their flap to prove to a coercer how they voted: the flap is used to identify batch k, then the bulletin board is used to verify the

batch tally is equal to the number of cast votes $T_{k,c} = |\mathbb{C}_k|$. If these conditions are met, then the voter must have voted for candidate c. This vulnerability may be exploited if it is sufficiently likely. We call batches vulnerable to this attack *uniform* batches, and say the election is *uniformity-free* if no batch is uniform.

Lemma 1. *Consider an election of N ballots and C candidates, where each candidate receives at most T^* votes, and N_{aud} ballots are audited/invalid. If voters are randomly divided into K batches of sizes $(N_k)_{k\in[K]}$ with $N_* \le N_k \le N^*$, then the election is uniformity-free with probability at least* $1 - KC\left(\frac{T^*+N_{\text{aud}}}{N}\right)^{N_*}$.

Proof. Let $\mathcal{U}_{k,c}$ denote the event that batch k contains only votes for candidate c. The number of votes for c follows a hypergeometric distribution so that $\Pr(\mathcal{U}_{k,c}) = \sum_{r=1}^{N_k} \binom{T_c}{r}\binom{N_{\text{aud}}}{N_k-r}/\binom{N}{N_k}$, where each summand is the probability that the batch contains r votes for candidate c, and $N_k - r$ audited/invalid ballots. Using Vandermonde's identity, $\Pr(\mathcal{U}_{k,c}) \le \binom{T_c+N_{\text{aud}}}{N_k}/\binom{N}{N_k} = (T_c + N_{\text{aud}})_{N_k}/(N)_{N_k}$, where $(n)_m$ is the *falling factorial*. Because the function $f(x) = \frac{T_c+N_{\text{aud}}-x}{N-x}$ is decreasing, the ratio of falling factorials is at most $\left(\frac{T_c+N_{\text{aud}}}{N}\right)^{N_k} \le \left(\frac{T_c+N_{\text{aud}}}{N}\right)^{N_*}$. The election fails to be uniformity-free if $\mathcal{U}_{k,c}$ occurs for any k or c, i.e. the event $\bigcup_{k\in[K]}\bigcup_{c\in[C]}\mathcal{U}_{k,c}$ occurs. Taking a union bound $\Pr(\cup_{k\in[K]}\cup_{c\in[C]}\mathcal{U}_{k,c}) \le \sum_{k\in[K]}\sum_{c\in[C]}\Pr(\mathcal{U}_{k,c})$ gives the desired result.

□

Collusion Attacks. A group of voters (*colluders*) may be able to work together to deduce the vote of a victim. If the colluders honestly share their own plaintext votes within the group, they can obtain the tallies for other ballots within their batches. If a batch k is complete on the bulletin board so that its tally $(T_{k,c})_{c\in[C]}$ can be extracted, then colluders can subtract their collective plaintext votes from each $T_{k,c}$ to find the tally of votes from batch k that *did not* come from the colluders. If there is only **one** ballot $i \in A^{-1}(k)$ that was not cast/audited by the colluders, they can work out the exact plaintext vote of ballot i.

If a batch contains at least **two** non-colluding voters, the remaining tally is an aggregate of multiple votes, mathematically upholding individual privacy [HZ06]. If a batch contains only colluders, then they cannot break the privacy of a victim. We call a batch *collusion-free* if it does not contain exactly one non-colluding voter, and we say the election is *collusion-resistant* if every batch is collusion-free.

Lemma 2. *Consider an election of N ballots where $\tilde{N}$ ballots come from colluders. If voters are randomly divided into K batches of sizes $(N_k)_{k\in[K]}$ with $N_* \le N_k \le N^*$, then the election is* collusion-resistant *with probability at least* $1 - KN^*\left(\frac{\tilde{N}}{N}\right)^{N_*-1}$.

Proof. Let C_k denote the event that batch k is *not* collusion-free, so that it contains exactly 1 colluder. The number of colluders in batch k follows a hypergeometric distribution, so we have $\Pr(C_k) = \binom{\tilde{N}}{N_k-1}\binom{N-\tilde{N}}{1}/\binom{N}{N_k} = N_k \frac{(\tilde{N})_{N_k-1}}{(N)_{N_k-1}}(1 - \frac{\tilde{N}-N_k+1}{N-N_k+1})$. Using the inequality of the ratio of falling factorials as in Lemma 1, $\Pr(C_k) \leq N_k(\frac{\tilde{N}}{N})^{N_k-1} \leq N^*(\frac{\tilde{N}}{N})^{N_*-1}$. The election fails to be collusion resistant if at least one batch is not collusion-free, i.e. the event $\bigcup_{k\in[K]} C_k$ occurs.. Taking a union bound gives the desired result.

□

If an election is both uniformity-free and collusion-free, we say it is *privacy-preserving*.

Theorem 3. *Consider an election of N ballots and C candidates, where each candidate receives at most T^* votes, N_{aud} ballots are audited/invalid, and $\tilde{N}$ ballots come from colluders. If voters are randomly split into K batches of sizes $(N_k)_{k\in[K]}$ with $N_* \leq N_k \leq N^*$, then the election is privacy-preserving with probability at least $1 - KN^*\left(\frac{\tilde{N}}{N}\right)^{N_*-1} - KC\left(\frac{T^*+N_{\text{aud}}}{N}\right)^{N_*}$.*

Proof. Let $\mathcal{P}_k$ be the event that batch k is *not* privacy-preserving. The election fails to be privacy-preserving if any batch is either not unanimity-free or not collusion-resistant. Then $\mathcal{P}_k = C_k \cup \bigcup_{c\in[C]} \mathcal{U}_{k,c}$. Applying a union bound to $\bigcup_{k\in[K]} \mathcal{P}_k$ and using the results in Lemma 1 and Lemma 2 give the desired result. □

For a particular election, let $P_{\text{priv}}(K) = 1 - KN^*(\frac{\tilde{N}}{N})^{N_*-1} - KC(\frac{T^*}{N})^{N_*}$ be the lower bound of the probability that the election is privacy-preserving that was established by Theorem 3. As a function of the number of batches K, $P_{\text{priv}}(K)$ is decreasing, so a smaller number of batches increases the likelihood of the election being privacy-preserving.

5.2 Missing Votes

It may happen that a ballot is not published to the bulletin board. These ballots, which we call *missing*, cause their batch to be *incomplete*, preventing the batch tally from being verified. Consequently, these batches cannot be included in the sample frame of batches during an audit, reducing the amount of verifiable information available. If too few ballots are in complete batches, the audit may fail by exhausting the sample frame before statistical confidence in the election outcome is attained. If a correct election outcome can be asserted at risk limit α by some RLA algorithm without performing a full hand count, we say the election is α-auditable.

We consider the $S \in [N]$ missing ballots to stem from *voter spoiling*: The ballot is damaged by the voter, preventing a bulletin board entry from being created when passed through a

scanner, which renders its batch incomplete. Ballots may be damaged in other ways due to mishandling by the EA, but we cannot model this due to a lack of available data.

Number of ballots in complete batches. Since ballots are assigned to batches randomly, we may assume that any two ballots are lost independently of each other with probability $p = S/N$. The event that a ballot is lost is therefore a Bernoulli trial with probability p. If batch $k \in [K]$ contains N_k ballots, then the number of missing ballots in batch k has binomial distribution with parameters(N_k, p). The probability that the batch is complete is then $(1-p)^{N_k}$.

Let X be a random variable that is the number of ballots contained in complete batches. Since batches are of approximately equal size N/K, the number of complete batches $\frac{XK}{N}$ has a binomial distribution with parameters $(K, (1-p)^{N/K})$.

If a batch-level RLA algorithm $\mathcal{R}$ at risk limit α samples batches containing x^* ballots, then we must have $X \geq x^*$ for the election to be α-auditable. We define $P_{\text{aud}}(K) = \Pr(X \geq x^*)$ as a function of K. It can be shown that $P_{\text{aud}}(K)$ is eventually increasing for fixed x^*.

This establishes the desired trade-off between an election being auditable and privacy-preserving: $P_{\text{aud}}(K)$ and $P_{\text{priv}}(K)$ have opposite monotonicities. This can also be realised intuitively: Large batches are less likely to be uniform or to contain a high number of colluders, but they are more likely to be incomplete since it only takes one ballot to go missing.

5.3 Scan Error

A scan error causes a ballot to be recorded contrary to the voter's intent. The model we adopt considers non-correlated scan errors, where each cast ballot independently has probability $f \in [0, 1)$ of being scanned incorrectly, such that it is counted as a vote for a uniformly random different candidate. One could also consider correlated scan errors, which may favour a particular candidate. For the most robust analysis, incorrectly scanned ballots should be treated as being cast by "evil zombies" that act to make the audit as inefficient as possible [BS12], which in our case would be a vote for the reported runner-up of the contest.

Scan errors may also manifest the scanning machine failing to read any information from a cast ballot, preventing it from being included in the count and causing an incomplete reported result. We do not consider these errors due to a lack of available data.

5.4 Data Analysis

The US Election Assistance Commission (EAC) provides comprehensive reports on US elections [Un], offering data on the number of missing ballots S. There is no systematic

reporting of in-person ballot spoiling, but rejected mail ballots must be accompanied by a rejection reason. We approximate p as the proportion of spoiled mail ballots.

Vote scanning in the United States is typically done by privately manufactured optical scanning machines [No10; Wa21]. It is difficult to obtain accurate information on scanning machine failures due to inaccurate or uncooperative audits [Ga09]. However, with transparent post-election audits becoming more common at a state level, results from previous audits can be used to estimate the proportion f of ballots that are scanned incorrectly and introduce a discrepancy between the VVPR and CVR. From the 15 states that conducted risk-limiting auditing during the 2024 presidential election, we use the result [Rh] that yielded the highest proportion of discrepancies between the CVR and auditor interpretation of a ballot, which we use as f.

Data on presidential and midterm elections in the United States published by the Election Assistance Commission (EAC) [Un] and Harvard Dataverse [DL17a; DL17b; DL17c] between 2016 and 2024 are used to simulate the division of ballots into batches, loss of ballots, and scanning of ballots for 220 elections.

For each election, numerical methods were used to determine whether the election was α-auditable with probability at least $1 - \alpha$ for $\alpha = 3\%$. If yes, then there exists a K (the number of batches) such that $P_{\text{aud}}(K) \geq 1 - \alpha$. We further enforce $K \leq \frac{N}{3}$: without this constraint, there may exist a batch containing at most two ballots, which trivially prevents the election from being privacy-preserving. If such a K exists, let $K_*(\alpha)$ denote the smallest K satisfying $P_{\text{aud}}(K) > 1 - \alpha$. For each contest admitting a K_*, $P_{\text{priv}}(K_*)$ was computed.

Of the 220 simulated elections for $\kappa = 1.2$, 196 (89%) admitted a value of $K_*(3\%)$. Of these 196 simulated elections admitting a K_*, 191 (97%) satisfied $P_{\text{priv}}(K_*) > 0.97$. In total, for 87% of the surveyed elections, there existed a choice of K such that the election was 3%-auditable and privacy-preserving with probability at least 97%.

The number of complete batches follows a binomial distribution with parameters $(K, (1 - p)^{N/K})$, so the expected proportion of *ballots* in complete batches is $(1 - p)^{N/K}$. Figure 3 shows the relationship curve between ballot missing rate p and expected proportion of ballots in complete batches for selected batch sizes N/K. For each election, a *BRAVO* [LSY12] RLA was simulated and the median proportion of sampled ballots y was recorded, along with the ballot missing rate x. The election is represented on Figure 3 as an $\times$ at (x, y). An $\times$ below a particular N/K curve indicates that the election is 3%-auditable with probability at least $\frac{1}{2}$. For batch size $N/K = 300$, 213 of the 220 elections (97%) were below the curve.

Table 1 shows the expected percentage of ballots in complete batches for selected ballot missing rates and batch sizes. If the ballot missing rate is at most 0.5%, then batch sizes of 200 will yield 36.7% of ballots in complete batches in expectation. This far exceeds the highest median proportion of ballots sampled (0.32%) during a *BRAVO* RLA across the analysed elections.

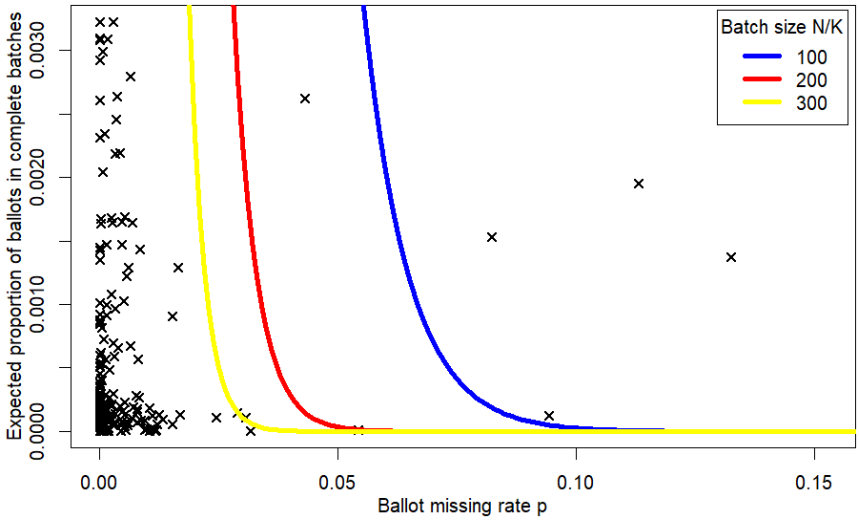


Fig. 3: Expected proportion of ballots in complete batches against ballot missing rate p; an $\times$ at position (x, y) represents an election with ballot missing rate x and median proportion of ballots sampled during a BRAVO [LSY12] RLA y.

	$\frac{N}{K} = 5$	$\frac{N}{K} = 10$	$\frac{N}{K} = 50$	$\frac{N}{K} = 100$	$\frac{N}{K} = 200$	$\frac{N}{K} = 300$
$p = 0.1\%$	99.5%	99.0%	95.1%	90.5%	81.9%	74.1%
$p = 0.5\%$	97.5%	95.1%	77.8%	60.6%	36.7%	22.2%
$p = 1\%$	95.1%	90.4%	60.5%	36.6%	13.4%	4.9%
$p = 5\%$	77.4%	59.9%	7.7%	0.6%	0.0%	0.0%

Tab. 1: Expected percentage of ballots in complete batches for selected ballot missing rates p and batch sizes $\frac{N}{K}$

6 Proof of Well-Formedness

The proofs of well-formedness ensure that each pre-computed $V_{i,c}$ on ballot i is a valid commitment for some candidate $j \in [C]$, consistent with the ballot's random commitments X_i, Y_i. Specifically, the committed vote must satisfy the condition that $(g, X_i, Y_i, V_{i,c}/g^{E(j)})$ forms a Diffie-Hellman tuple for at least one $j \in [C]$. That is $\exists j \in [C] : \log_g Y_i = \log_{X_i}\left(\frac{V_{i,c}}{g^{E(j)}}\right)$ where $E(j)$ is the public encoding for candidate j. To ensure universal verifiability, the Election Authority (EA) must prove $V_{i,c}$ belongs to the set of valid commitments without revealing *which* candidate it encodes. We achieve this using a Non-Interactive Zero-Knowledge (NIZK) proof for the disjunction of equalities of discrete logarithms [CDS94]. The prover demonstrates knowledge of the witness y_i and an index $c \in [C]$ satisfying the relation. Committing to x_i and y_i directly via $X_i = g^{x_i}$ and $Y_i = g^{y_i}$ prevents a malicious EA

from altering the blinding factors to covertly encode a different vote. This proof uses the Chaum-Pedersen protocol [CP92] transformed into a 1-out-of-C "OR" proof via the Cramer-Damgård-Schoenmakers (CDS) technique [CDS94], made non-interactive using the Fiat-Shamir heuristic [FS86]. Let $\mathcal{L}_j$ represent the discrete logarithm relation for candidate branch j: $\log_g Y_i \stackrel{?}{=} \log_{X_i}(V_{i,c}/g^{E(j)})$. The prover aims to show $\mathcal{L}_c$ holds for the true printed candidate c. The proof $\pi_{i,c}$ consists of the tuple $(\mathbf{k}, \mathbf{r})$, where $\mathbf{k} = \{k_1, \ldots, k_C\}$ are challenge shares and $\mathbf{r} = \{r_1, \ldots, r_C\}$ are responses. To verify the proof, anyone can perform the checks for reconstructing the commitments and challenge consistency that are $\widetilde{T}_j = g^{r_j} Y_i^{k_j}$, $\widetilde{T}'_j = X_i^{r_j} \left(\frac{V_{i,c}}{g^{E(j)}}\right)^{k_j}$ and $\sum_{j\in[C]} k_j \stackrel{?}{\equiv} H\left(g, X_i, Y_i, V_{i,c}, \{\widetilde{T}_j, \widetilde{T}'_j\}_{j\in[C]}\right) \pmod q$. Because our construction is a direct application of the CDS transformation to the Chaum-Pedersen protocol, it inherits its security properties [CDS94].

Theorem 4. *The proof $\pi_{i,c}$ consists of tuple $(\mathbf{k}, \mathbf{r})$, is a secure NIZK proof of knowledge [CDS94] satisfying Completeness, Special Soundness, and Honest-Verifier Zero-Knowledge (HVZK) in the Random Oracle Model.*

Sketch. Completeness follows by construction; an honest prover generates the true branch commitments using a uniform random nonce $t \in \mathbb{Z}_q$, Letting k denote the output of the hash function H, the prover calculates the true challenge share as $k_c = k - \sum_{j\neq c} k_j \pmod q$, and sets the response $r_c = t - y_i k_c \pmod q$. The false branches are simulated perfectly. *Special Soundness* is guaranteed because any two accepting transcripts with identical initial commitments but distinct global challenges ($k \neq \hat{k}$) must differ in at least one challenge share $k_\ell \neq \hat{k}_\ell$. This allows a standard extractor to uniquely recover the witness: $y_i = (\hat{r}_\ell - r_\ell)/(k_\ell - \hat{k}_\ell)$. Finally, *Zero-Knowledge* is achieved by a standard simulator that picks all responses r_j and challenge shares k_j uniformly at random (subject to the summation constraint), computes the commitments backwards, and programs the random oracle H to output the sum. This produces a transcript statistically indistinguishable from a real proof without knowledge of the witness y_i or the index c. □

References

[Be06] Benaloh, J.: Simple Verifiable Elections. In: EVT. 2006.

[Be11] Benaloh, J. et al.: {SOBA}: Secrecy-preserving Observable Ballot-level Audit. In: EVT/WOTE 11. 2011.

[BS12] Banuelos, J. H.; Stark, P. B.: Limiting risk by turning manifest phantoms into evil zombies. arXiv preprint arXiv:1207.3413, 2012.

[BST19] Benaloh, J.; Stark, P. B.; Teague, V.: VAULT: Verifiable audits using limited transparency. Proceedings of E-Vote ID, 2019.

[BT00] Blackburn, S. R.; Teske, E.: Baby-step giant-step algorithms for non-uniform distributions. In: International Algorithmic Number Theory Symposium. Springer, pp. 153–168, 2000.

[CDS94] Cramer, R.; Damgård, I.; Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: CRYPTO. Pp. 174–187, 1994.

[CP92] Chaum, D.; Pedersen, T. P.: Wallet databases with observers. In: Annual international cryptology conference. Springer, pp. 89–105, 1992.

[Cr96] Cramer, R. et al.: Multi-authority secret-ballot elections with linear work. In: CRYPTO. Pp. 72–83, 1996.

[DKR06] Delaune, S.; Kremer, S.; Ryan, M.: Coercion-resistance and receipt-freeness in electronic voting. In: IEEE CSFW. 12–pp, 2006.

[DL17a] Data, M. E.; Lab, S.: U.S. House 1976–2024, version V15, 2017, https://doi.org/10.7910/DVN/IG0UN2.

[DL17b] Data, M. E.; Lab, S.: U.S. President 1976–2020, version V8, 2017, https://doi.org/10.7910/DVN/42MVDX.

[DL17c] Data, M. E.; Lab, S.: U.S. Senate statewide 1976–2020, version V7, 2017, https://doi.org/10.7910/DVN/PEJ5QU.

[FS86] Fiat, A.; Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Euocrypt. Pp. 186–194, 1986.

[Ga09] Garber, M. K.: Examining Florida's High Invalid Vote Rate in the 2008 General Election. 2009.

[Go] Goodman, S. et al.: Risk-Limiting Post-Election Audits: Why and How.

[Ha14] Hao, F. et al.: Every Vote Counts: Ensuring Integrity in Large-Scale Electronic Voting. USENIX Journal of Election Technology and Systems (JETS) (3), pp. 1–25, 2014.

[HZ06] Hao, F.; Zieliński, P.: A 2-round anonymous veto protocol. In: SPW. 2006.

[LS12] Lindeman, M.; Stark, P. B.: A gentle introduction to risk-limiting audits. IEEE Security & Privacy 10 (5), pp. 42–49, 2012.

[LSY12] Lindeman, M.; Stark, P. B.; Yates, V. S.: BRAVO: Ballot-polling Risk-limiting Audits to Verify Outcomes. EVT/WOTE 125, p. 129, 2012.

[No10] Norden, L. D.: voting system failures: a database solution. Brennan Center for Justice, 2010.

[Rh] Rhode Island Board of Elections: Rhode Island Risk Limiting Audit Center, https://elections.ri.gov/elections/risk-limiting-audit-center.

[St08] Stark, P. B.: Risk-limiting post-election audits. 2008.

[St20] Stark, P. B.: Sets of half-average nulls generate risk-limiting audits: SHANGRLA. In: International conference on financial cryptography and data security. Springer, pp. 319–336, 2020.

[Un] United States Electoral Assistance Commission: Election Administration and Voting Survey (EAVS) Comprehensive Report, https://www.eac.gov/research-and-data/studies-and-reports.

[Wa21] Wagner, P.: Election Security: Technology, Risks, and Mitigations. Risks, and Mitigations (February 23, 2021), 2021.

[Wa92] Wald, A.: Sequential tests of statistical hypotheses. In: Breakthroughs in statistics: Foundations and basic theory. Springer, pp. 256–298, 1992.