VAULT: Verifiable Audits Using Limited Transparency

Josh Benaloh[*1], Philip B. Stark[2][0000−0002−3771−9604], and Vanessa Teague[3][0000−0003−2648−2565]

[1] Microsoft Research benaloh@microsoft.com
[2] University of California, Berkeley stark@stat.berkeley.edu
[3] University of Melbourne vjteague@unimelb.edu.au

Abstract. Risk-limiting audits (RLAs) can provide strong evidence that reported election outcomes are correct, on the assumption that the paper trail of voter-verified ballots is trustworthy. Ballot-comparison RLAs involve comparing a human reading of voter intent from the paper ballot to the voting system's electronic representation of voter intent for that ballot, the cast-vote record (CVR). Ballot-comparison RLAs first check that the full list of CVRs reproduces the reported results, then compare manual readings to CVRs for randomly selected ballots. For a ballot-comparison RLA to deserve public trust, the public must be able to validate those two steps. The easiest way to do that is to publish the entire list of CVRs. However, if every CVR is published, "Italian attacks" via pattern voting can be used to coerce voters or to facilitate selling votes.

Keywords: risk-limiting audit · homomorphic encryption · elections

1 Introduction

Over the last decade, *risk-limiting audits* (RLAs) [19,12] have gained traction as a method for verifying whether reported election outcomes[4] accurately reflect the underlying paper trail. A recent report of the National Academies of Science, Engineering, and Medicine [16] advocates RLAs. They are performed routinely in Colorado, and are mandated by law now in Colorado, Rhode Island, Virginia, and Texas. There have been about 40 pilot audits in California, Colorado, Indiana, Michigan, New Jersey, Ohio, Rhode Island, Virginia and in Denmark.

RLAs involve manually examining random samples of paper ballots. If and when the sample provides adequately strong evidence that the reported outcome is correct, the audit stops; otherwise, it progresses to a full manual tally to set the record straight.

* Authors listed alphabetically.

[4] *Outcome* means the political outcome—the candidate(s) or position(s) that won—not the exact vote counts.

However, auditing using rigorous statistical criteria is not enough to justify public confidence in election outcomes. An audit should not only allow insiders or approved auditors to check the results, it should also provide the public with enough information to verify that the audit was conducted properly and did not stop prematurely. At the same time, the public information should not compromise voter privacy. When RLAs are considered as a public verification process, their requirements closely resemble the *public verifiability* property of end-to-end verifiable elections.

The most efficient kinds of RLAs require a commitment to the interpretation of each ballot in advance of the audit. Traditionally, this commitment is made by producing a complete, plaintext statement of the contents of each ballot. Unfortunately, this can introduce a privacy problem for some election types. In this paper we show how a cryptographic commitment can be used as the basis of an RLA with essentially the same public verifiability as a traditional plaintext statement, but much better protection of individual vote privacy.

The methods are immediately useful in California, Colorado, Rhode Island (USA) and New South Wales (Australia).

We first describe how ballot-comparison risk-limiting audits work, then explain the privacy problem we are solving (Section 1.2) and the cryptographic tools we can use instead of plaintext commitments. Section 2 outlines the main advantages and shortcomings of VAULT compared with prior art. Section 3 then gives an overview of current audit law and practice in some example jurisdictions. The technical details of our approach are explained in Section 4, with some detailed examples in Section 5 and an informal argument for its main security properties in Section 6.

1.1 Ballot-comparison risk-limiting audits

Unlike traditional post-election audits, RLAs adjust the sample size to attain a desired level of confidence that electoral outcomes are correct, given what the audit finds as it progresses. There are many methods for conducting RLAs. The most efficient, measured by the number of ballots that need to be inspected when reported outcomes are correct, is a *ballot-comparison audit*. Ballot-comparison audits are possible only if the vote tabulation system creates an electronic interpretation of voter's preferences for each ballot—a *cast vote record* (CVR)—in such a way that the corresponding paper ballot is uniquely identified and can be retrieved for manual inspection by auditors, so that their interpretation of the ballot can be compared to the CVR.

Existing protocols for ballot-comparison RLAs start with:

1. A *ballot manifest*, which describes in detail how the physical ballots are stored, so that ballots can be selected randomly and retrieved.
2. A *commitment* by the voting system to the full set of CVRs.[5]

[5] Here, *commitment* is a term of art. It means that something about the CVRs must be published in such a way that observers can tell whether the CVRs that the audits

To conduct the audit, auditors first confirm that applying the social choice function to those CVRs yields the reported results, and that there are not more CVRs than ballots.[6] (The social choice function is the rule for figuring out who won, such as plurality, multi-winner plurality, majority, IRV, or D'Hondt.) The audit proceeds by randomly selecting ballots and checking whether the corresponding CVRs match a human reading of the paper.

Ballot-comparison audits are like checking an itemized expense report using paper receipts. The first step is to check whether the itemized expenses add up to the total requested, and whether there is a receipt for every item. The second step is to spot-check the amounts of the reported expenses against the amounts listed on the receipts. Requiring the traveler to itemize expenses keeps the traveller from being able to fudge the numbers after the fact. Checking whether the itemized expenses add up to the requested reimbursement prevents a traveler from reporting every receipt accurately, but adding the expenses incorrectly.

Analogously, requiring a commitment to the CVRs before the audit starts keeps the system from simply generating CVRs that match whatever ballots the audit selects; and verifying that the collection of commitments imply the reported electoral outcomes ensures that if the commitments accurately reflect their corresponding ballots, the reported electoral outcomes must be correct.

A public auditing algorithm would therefore consist of:

1. Checking that the social choice function, applied to the CVRs, does indeed produce the announced election result.
2. Checking that the Risk Limiting Audit has been properly applied to the CVRs and paper ballots. This includes verifying that the random ballot selections are properly computed, checking that the correct paper ballot is retrieved according to the ballot manifest, applying the RLA risk computation to the ballot's true value, and checking that the audit stops only when the RLA instructs it to (or falls back to a full manual recount).

In this work, we assume that VAULT takes as input a valid ballot-comparison RLA algorithm and concentrate only on the use of cryptographic rather than plaintext commitments. Important details such as how to verify that the ballots are properly selected at random, are out of scope.

1.2 Public evidence and voter privacy

Ballot-comparison RLAs provide strong public evidence that reported outcomes are correct if the commitment to the CVRs is public, if the ballot selection process is publicly verifiable, and the public can observe whether the selected ballots match the commitments about the corresponding CVRs.

---

check against the ballots are altered during the audit. One way to commit to the CVRs is simply to publish them all.

[6] There are conservative methods for dealing with a mismatch between the number of CVRs and the number of ballots in the ballot manifest; see [2].

However, committing to the full set of CVRs by publishing them all may compromise the anonymity of the vote and enable an attacker to coerce voters through "pattern voting." For instance, suppose an employer is running for mayor and wants to ensure getting the vote of all employees. The employer can select a lesser office on the ballot (e.g., "dog-catcher") and threaten that each employee who wants to remain employed should cast a vote with the employer selected for mayor and the employee's own name written in for dog-catcher. When the CVRs are released, the employer can check which employees complied with the demand. Even if write-in votes are not possible, the employer could select, for each employee, a unique pattern of votes on "downballot" contests and then check whether the patterns show up in the published CVRs. Complex voting systems such as Range Voting and Instant Runoff Voting (IRV) are susceptible even when there is only one race on the ballot.

1.3 Cryptographic commitments and homomorphic tallying

Here we show that cryptography provides an alternative way to commit publicly to the CVRs. This *cryptographic commitment* still lets the public check whether—on the assumption that the cryptographic commitments accurately reflect the votes on the underlying ballots—the reported results are correct, and also lets audit observers check (statistically) whether the commitments were accurate enough that the reported outcome is correct. Using appropriately designed cryptographic commitments protects voter privacy while still allowing the public to verify the audit.

Effective verification, of course, depends upon the protocol being sound. A verification mechanism may seem to be secure but actually leave gaps that make an election's results unverifiable. For instance, part of the protocol may require the system to prove that the commitment for a CVR does not hide a negative vote, or more than one vote for a particular candidate. If the system could fake a proof that the committed value was valid, it could fake election results and evade detection with probability much higher than the RLA's risk limit. This is not merely hypothetical: the protocol for the Scytl/SwissPost Internet voting system[7] contains just such a flaw (Lewis, Pereira, and Teague 2019).

The remainder of this paper describes how techniques that for decades have been used in end-to-end verifiable (E2E-V) systems can be re-purposed to enable publicly verifiable ballot-comparison RLAs without revealing the contents of ballots other than those selected at random in the audit.

We use a cryptographic commitment scheme and denote by $c = E(m, r)$ the commitment to message m with randomness r. The commitment is *opened* when the committer produces (m, r), thus allowing anyone to check that $E(m, r) = c$. The scheme must be both *hiding* and *binding*, meaning that the commitment does not reveal the message, and that it is infeasible to open a commitment in

[7] The flaw also affects the iVote Internet voting system deployed in New South Wales, Australia.

two different ways, *i.e.* to find (m, r) and (m', r') s.t. $m \neq m'$ but $E(m', r') = E(m, r)$. Precise definitions can be found in any cryptography textbook [5,10].

E must also satisfy the homomorphic addition property:

$$E(m_1, r_1).E(m_2, r_2) = E(m_1 + m_2, r_1 \oplus r_2)$$

where . and $\oplus$ are easily-computed functions, usually modular multiplication and addition. For example, E might be an El Gamal encryption putting the message in the exponent. This is perfectly binding (because there is only one possible decryption of any ciphertext), but only computationally hiding (because an attacker who guesses the key can compute the plaintext). Alternatively, we could use Pedersen commitments [18], which are perfectly hiding but only computationally binding.

Using the homomorphic property, committed values can be combined by any observer to form a commitment to the sum of those values. The committed value can then be publicly opened, so anyone can verify that the claimed total is correct. With homomorphic tallying, individual votes are never decrypted or revealed.

Homomorphic tallying has been used in numerous cryptographic voting protocols to enable independent verification that a set of encrypted votes corresponds to an announced tally, without revealing the contents of individual ballots.

Some systems use perfectly hiding cryptographic commitments to achieve *everlasting privacy* [14,15,9,1], meaning that the published data does not expose information about the individual ballot even to an attacker with unlimited computational power. VAULT can be implemented with perfectly hiding cryptographic commitments and hence provide everlasting privacy for those ballots that are not audited.

The key contribution of this paper is the observation that homomorphic tallying also makes it possible to conduct ballot-comparison audits without revealing the contents of any ballots other than those selected at random in the audit, which is generally a small fraction of the ballots that were cast.

2 Related Work and VAULT's advantages and limitations

2.1 SOBA

SOBA (*secrecy-preserving observable ballot-level audits* [3] addresses the coercion problem by splitting ballots into their constituent votes and then creating a complex web of hash commitments that can be used to verify the required ballot properties without publishing full ballots. While SOBA is effective, it is complicated and unintuitive. No jurisdiction has used it to alleviate the real and practical problem of enabling public verification of ballot-comparison audits without putting voter privacy at risk.

SOBA and VAULT both rely heavily on cryptographic commitments, but in different ways. Public perception of the methods might be quite different, as a

result. In SOBA, people need to trust the cryptographic commitments to ensure that the plaintext votes for different contests really do correspond to "slicing" ballots into separate contests. The commitments prevent a cheating authority from reassembling sliced ballots any way they like—but the public can tally the plaintext votes themselves.

For VAULT, the public must rely on homomorphic encryption to check whether the commitments imply the reported outcomes. In SOBA it is less obvious that integrity relies on the cryptography, so SOBA may engender more public trust even though it relies just as essentially on cryptographic commitments.

Also, SOBA works by splitting up a ballot, which solves the problem for some social choice functions (such as Borda Count or Condorcet methods), and for US-style ballots with multiple questions on one ballot paper. It does not extend in an obvious way to Instant Runoff Voting (IRV), in which one vote may contain enough information for coercion, but it can not be divided into smaller parts while still allowing the social choice function to be computed.

Here we show how privacy-preserving ballot-comparison audits can be conducted far more simply and convincingly.

2.2 End-to-End Verifiable Elections

E2E-verifiability is generally achieved by publishing an encryption of all votes recorded in an election. An election is then *end-to-end (E2E) verifiable* if two properties are satisfied.

- Voters can confirm that their own votes have been correctly recorded.
- Voters and observers can confirm that all recorded votes have been correctly tallied.

The first of these properties is often referred to as *individual verifiability* while the second is typically known as *universal verifiability*. It is the universal verifiability property that is of interest for RLAs, because it closely matches the properties required of CVRs in a publicly observable ballot-comparison audit. However, there are some important differences.

The primary difference between VAULT and E2E-V is the level of protection that needs to be afforded to the raw data. In E2E-verifiability, releasing even a single raw ballot can directly compromise a voter's privacy, because each voter in an E2E-verifiable election receives a receipt tied to the voter's encrypted ballot. To prevent rogue individuals from decrypting the CVRs, decryption keys used for E2E-verifiability are typically shared amongst multiple independent parties in a way that some subset must cooperate to decrypt anything.

In contrast, for a ballot-comparison audit, the electoral process is assumed to have already done something to disassociate ballots from the identities of the voters who cast them. Thus the threat is lower: releasing an individual CVR does not immediately compromise privacy because ballots and CVRs are not linked to individual voters. Ballot-comparison audits require unsealing individual CVRs

as the audit progresses. It is inefficient to require a quorum of keyholders to convene and execute a decryption protocol every time an RLA selects additional ballots. It is therefore both desirable and sufficient for the encrypted CVRs to have a single decryption key—presumably held by election administrators.

2.3 Effectiveness of VAULT's coercion-resistance

While it might appear as though the release of the complete set of votes on even a single ballot creates a privacy risk, the true risk comes from the release of the contents of *most or all* ballots. In order for effective coercion to take place, there needs to be a means by which a coercer can determine that a coerced voter *did not* vote as prescribed. However, if the contents of only a minority of ballots are revealed (at random), a coerced voter can simply assert that the voter's ballot was not among those that were revealed.

While the new approach thwarts coercion, it is less effective against voluntary vote buying and vote selling. Even when the contents of only a small fraction of ballots are released, a lottery bounty might effectively purchase votes. A voter who might sell a vote for, say, $10 might be just as willing to sell a vote for, say, a 1% chance of getting $1,000. A vote buyer could therefore assign patterns to individual voters and pay a large bounty to any voter whose assigned pattern appears in a released CVR. A vote buyer's potential payout could even be protected by tying the size of the bounty to the number of ballots whose contents are revealed.

3 Current audit law and practice

3.1 Colorado

Colorado counties that perform ballot-comparison audits upload ballot manifests and CVRs to state-provided, open-source software called RLATool. The Secretary of State publishes a cryptographic hash of the entire CVR file,[8] but not individual plain text CVRs. The officials who audit the paper ballots manually and enter their reading of voter intent into RLATool generally do not have access to the CVRs, and do not calculate whether there is a discrepancy between the CVR and their interpretation: that is calculated by RLATool. Members of the public do not have access to the CVRs, before, during, or after the audit. After each round of the audit, the state generates a report that lists each ballot inspected and whether or not the CVR had a discrepancy, contest by contest.[9] The public currently has no way to check whether the comparison was done correctly.

[8] See, e.g., https://www.sos.state.co.us/pubs/elections/RLA/files/2018G/round_1/cvr_hash.csv (last visited 15 May 2019).

[9] See, e.g., https://www.sos.state.co.us/pubs/elections/auditCenter.html (last visited 15 May 2019).

3.2 California

California AB2125, signed into law in 2018,[10], authorizes pilot RLAs in 2020. Section 15366(b) defines *ballot-level comparison audits* (i.e., ballot-comparison audits):

(1) The elections official uses an independent system to verify that the cast vote records created by the voting system or ballots created independent from the tally or ballot marking system yield the same election results as those reported by the voting system.

(2) The elections official compares some or all of those cast vote records to a hand-to-eye, human interpretation of voter markings from the corresponding ballot marked by the voter or the voter verified paper audit trail, as defined by Section 19271.

Section 15367(b)(2)(G) requires the Secretary of State to establish regulations so that "the audit process is observable and verifiable by the public." We interpret 15366(b)(1) in conjunction with 15367(b)(2)(G) to mean that the regulations must allow the public to verify that the CVRs used in ballot-comparison audits yield the reported results, and that the correct CVR is compared to each ballot selected for audit. That could be accomplished by publishing the entire set of CVRs in plain text—which could compromise voter privacy and facilitate vote-selling and coercion, as discussed above. Hence, it would be preferable to provide the public a way to ensure that the CVRs used in the audit yield the reported results without revealing every CVR. The approach we develop here solves the problem.

Constraints of Existing California Voting Systems

Conversations with California elections officials lead us to expect that the counties most likely to participate in the pilot have voting systems that produce CVRs that can be matched to ballots by relying on the order in which ballots are scanned, and that those counties are more likely to pilot ballot-comparison RLAs than ballot-polling RLAs.

Ballot imprinters were recently certified by the California Secretary of State; at least one jurisdiction likely to conduct audits under AB2125 plans to purchase imprinters. However, voting systems in most California counties cannot imprint

[10] https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB2125, last visited 18 April 2019

identifiers or salts on ballots.[11] To our knowledge, no current voting system in California can print salts on ballots.

Thus, to comply with AB2125 and still protect voter privacy to the maximum extent possible, a method that does not rely on imprinting salts on ballots is needed.

3.3 Australia

Australian federal elections, and some state elections, use an automated scanning process to digitize paper ballots before counting, but currently no law requires any auditing of any paper records at all. As far as we know there is no public auditing in practice either.

4 Technical Details

For a simple plurality election, the process of proving that a set of encrypted ballots corresponds to an announced tally is identical to what is done for E2E-verifiability. However, by generalizing this approach, we can accommodate RLAs for a broader class of elections including instant-runoff voting.

Assume a ballot manifest, known to the electoral authority but not publicly released, providing a unique ID number for each ballot and attesting to its contents.

Suppose we have a set of asserted tallies $\mathcal{A} = \{A_1, A_2, \ldots, A_J\}$ for the election. An assertion claims something about numbers that can be derived from the ballots, for example that a certain candidate's tally has some particular value. These each contribute to some (perhaps several) null hypotheses $\mathcal{N} = \{N_1, N_2, \ldots, N_I\}$ to be examined by RLA. Each paper ballot contributes some numerical value (most often 1, -1 or 0).

For example, if the election consists of a simple plurality election, then each assertion A_j might be the announced total of candidate j, and a_{ij} might be 1 if the ballot i is a vote for candidate j, zero otherwise. N_1 might be the hypothesis that a certain losing candidate actually got a higher tally (according to the paper ballots) than the announced winner. See Section 5 for detailed examples.

Note that some asserted tallies might be wrong though their dependent null hypotheses might still be demonstrably false—a small number of misrecorded votes, and hence some small errors in the announced tallies, don't usually alter the election result.

[11] Experience in Colorado shows that printing sequential identifiers on ballots substantially increases the speed and accuracy of retrieving ballots. The Humboldt County Elections Transparency Project does imprint the ballots before re-scanning the ballots using an independent, unofficial system. https://electionstransparencyproject.org/ While imprinting and rescanning could be the basis of a ballot-comparison *transitive* RLA of the kind conducted in pilots in California and Colorado, we do not anticipate that any California jurisdiction will attempt such an audit in 2020.

Let n be the total number of ballots. The audit proceeds as follows.

For each ballot b_i, for each assertion A_j, the EA posts a commitment to a_{ij}, which is a number representing b_i's contribution to A_j. This commitment is denoted by C_{ij}.

$$C_{ij} = E(a_{ij}, r_{ij}) \text{ where } r_{ij} \text{ is randomly chosen.}$$

Then for each assertion A_j, the sum of the contributions of all b_i's are computed (which is a public operation) and opened (which the authority has sufficient information to do, having produced the summands). That is,

$$C_j = \Pi_{i=1}^{n} C_{ij}$$

and the EA publishes $\Sigma_{i=1}^{n} a_{ij}$ and $\bigoplus_{i=1}^{n} r_{ij}$. This opening can be immediately checked.

The audit consists of randomly selecting a paper ballot b_i, locating its electronic record, and then for each assertion A_j $(j = 1 \ldots J)$, opening the commitments C_{ij}, by publishing the pair (a_{ij}, r_{ij}). This allows observers to check the commitment opening and verify that the committed values a_{ij} $(j = 1, \ldots, J)$ correctly describe ballot b_i's contributions to each assertion.

Each committed value a_{ij} is expected to fall within some set S_j of valid entries, defined at the beginning of the election. For example, in a standard first-past-the post election the set of valid contributions to a candidate's tally is $\{0, 1\}$; in a Borda election it is $\{0, 1, \ldots, n-1\}$. The RLA defines the assumed sets of expected values for each assertion, and the EA proves that each committed value is within its corresponding set. *It is critically important that the proven ranges match the RLA's assumptions.* We will denote the proof that a commitment c contains a value in set S as $ZKP_S(c)$. Depending on the set, these could be instantiated as witness-indistinguishable disjunctive proofs (Cramer, Damgård, and Schoenmakers 1994), range proofs (Mao 1998), (Camenisch, Chaabouni, and others 2008), (Bünz et al. 2017), etc.

The first step is for the EA to define the assertions, RLA algorithm and corresponding sets of valid committed values. This is shown in Algorithm 1. The idea is that the assertions form the set of facts to be audited—it is up to the public to verify that their conjunction implies the announced election outcome. More precisely, the set $\mathcal{N}$ of null hypotheses should obviously, when eliminated, imply that the announced election outcome is true, and the list of asserted tallies $\mathcal{A}$ should (if true) imply that all the null hypotheses are false.

The commitment process is shown in Algorithm 2. There, $r_{ij} \leftarrow R$ means that r_{ij} is chosen randomly and uniformly from set R.

Verification is Algorithm 3. If there are some committed votes that do not have corresponding paper ballots, this can be dealt with using the phantom/zombie approach of [2].

Algorithm 1 Election outcome statement–EA

Input: Election outcome; social choice function; Risk Limiting Audit algorithm $\mathcal{RLA}$.

1: Announce the election outcome
2: Define the set $\mathcal{N}$ of null hypotheses to be examined by RLA.
3: Define each assertion A_j for $j = 1..J$
4: **for** j=1..J **do**
5: Define the set S_j of valid single-ballot contributions to A_j.

Algorithm 2 Commitment and opening algorithm–EA

Input: Ballot manifest; election outcome statement; commitment algorithm E with randomness range R; set inclusion proof NIZKP ZKP.

1: **for** each ballot b_i **do** ▷ Make Commitments
2: **for** each assertion A_j **do**
3: $a_{ij} = b_i$'s contribution to assertion A_j
4: $r_{ij} \leftarrow R$
5: publish $C_{ij} = E(a_{ij}, r_{ij})$
6: publish $ZKP_{S_j}(C_{ij})$
7: **for** each assertion A_j **do**
8: compute $C_j = \Pi_i C_{ij}$ ▷ Aggregate commitments
9: publish $\Sigma_i a_{ij}$ and $\bigoplus_i r_{ij}$ ▷ Open the aggregate commitment
10: When ballot b_i is audited ▷ Auditing
11: Publish a_{ij}, r_{ij} for $j = 1, \ldots, J$
12: *Note: actually it is necessary to open the commitments only for those assertions for which the audit has not terminated.*

Algorithm 3 Commitment and opening verification algorithm—public

Input: EA's election outcome statement; audited paper ballots; Risk Limiting Audit algorithm $\mathcal{RLA}$; Commitment algorithm E; set inclusion proof verification algorithm.

Check that the conjunction of $\{A_j\}$ over all j
implies all the null hypotheses $\mathcal{N}$ are false.
Check that if all $\mathcal{N}$ are false, this implies that the announced election outcome is true.
If either of these checks fail, STOP and perform a full manual recount.
for each assertion A_j and each ballot b_i **do**
    checking that S_j matches the assumed set in $\mathcal{RLA}$.
    verify $ZKP_{S_j}(C_{ij})$.
for each assertion A_j **do** ▷ COMMITMENT VERIFICATION

    If the EA does not open C_j, STOP and conduct a full manual recount.
    Recompute C_j and check that $\Sigma_i a_{ij}$, $\bigoplus_i r_{ij}$ is a valid opening
    Check that $A_j = \Sigma_i a_{ij}$
for each ballot b_i that is audited **do** ▷ AUDITING VERIFICATION
    for $j = 1, \ldots, J$ **do**,
        verify that
        a_{ij}, r_{ij} is a valid opening of C_{ij} and
        a_{ij} accurately describes the paper ballot
        if the commitment opening is invalid or absent **then**
            if r_{ij} makes a valid opening of C_{ij} for some other value $a'_{ij} \in S_j$ **then**
                follow $\mathcal{RLA}$, with a'_{ij} as the apparent vote and the physical ballot as the true one.
            else
                follow $\mathcal{RLA}$, making the worst-case assumption about a_{ij}.
        if a_{ij} differs from the paper ballot **then**
            follow $\mathcal{R}$, with a_{ij} as the apparent vote and the paper ballot as the true one.

4.1 Defining the worst-case assumption

If the EA refuses (or is unable) to open a commitment, C_{ij}, or if a commitment opening doesn't verify, we must make the worst-case assumption about the message that was committed to. The worst-case assumption about a_{ij} is defined by the audit method and the valid set S_j. It might be different for each null hypothesis being tested.

Suppose for example that A_j declares a tally for some announced loser c_j, and that c_1 is the announced winner, in a single-winner plurality contest, with a tally announced by A_1. Then $S_j = \{0, 1\}$. Suppose we have retrieved some particular ballot b_i and observed its contents, but the EA refuses (or is unable) to open the commitment C_{ij}. The commitment must have contributed to A_j's homomorphic tally some value in the set S_j. Consider the implications for a particular RLA testing a particular null hypotheses N_k, which states that A_1 is a tally lower than or equal to A_j. The worst case assumption about a_{ij} is the maximum, over all values in S_j, of the discrepancy in favour of the announced winner compared with the true value on ballot b_i. This is one if b_i contains a vote for c_j, and zero otherwise. (The worst case is that a true vote for a loser was instead tallied as zero.) If the EA also refuses to open the commitment to a_{i1}, then a similar analysis shows that the worst case interpretation is another 1 if b_i shows a vote for the announced loser—if both of these happen, the RLA treats it as a two-vote overstatement.

The case in which the EA refuses to open the commitment might be ameliorated by using encryption (rather than other kinds of commitments) because then there is some set of authorities who hold the decryption keys, and may therefore open the commitment without having generated it. These authorities may still refuse to decrypt the message, however, so there still needs to be a way of incorporating this refusal into the audit.

For more expressive voting schemes such as Range Voting, if we let $A_j(b_i)$ be b_i's numerical contribution to assertion A_j, then the worst-case assumption for the discrepancy is $d_{worst_k} = \text{argmax}_{s \in S_j}\{s - A_j(b_i)\}$.

Note that the worst-case assumptions are chosen independently across different assertions. We never prove or check that the commitments about a single ballot are consistent—a cheating authority could have made various assertions about a ballot that are not consistent with any real ballot.

The above is sufficient data to conduct a Risk Limiting Audit, which must be paramaterised s.t. the set of possible committed values corresponds to S_j for each assertion A_j.

4.2 Putting it together with an RLA

We now have all the ingredients necessary to conduct a Risk Limiting Audit of the announced outcome, by testing the null hypotheses associated with each assertion.

The basis for RLAs described by [19] is simply to test a hypothesis about the mean of a finite non-negative population—in our case, we are testing the

hypothesis that the discrepancies between the paper ballot data and the committed values are large enough to alter the outcome. The set membership proofs guarantee that each individual discrepancy is bounded by a known value (it might be negative, but it is bounded below). Hence the statistics of the RLA work exactly as they would do in a traditional open-CVR-based audit with the same parameters.

4.3 Locating ballots and keeping track of salts

There are several different ways of doing the bookkeeping necessary to implement the algorithm.

1. The random openings r_{ij} could be printed directly on the paper ballots – either in plaintext or encrypted.[12]
2. In the case of multiple commitments per ballot, the random openings could be generated from a cryptographic PRNG, for which the seed was printed directly on the paper ballot.
3. The random openings could be posted, encrypted, on the WBB.
4. The index i could be printed on ballot b_i.
5. There could be no printing on the ballots, but they could be stored in a way that made the index associated with each ballot obvious to an observer.

When the only option available is 5, it is important to ensure a publicly-verifiable correspondence between the ballot IDs and their paper ballots. This protects against substitution of ballots during the audit. Accidental errors of this kind have caused problems during audits (Ottoboni 2019)—deliberate substitution could render the audit meaningless. Printing either ballot IDs (Option 4) or random commitment openings (Options 1 and 2) conveniently prevents this substitution, assuming that observers can see that all the ballots have been printed in advance.

Whether the IDs or the random openings are printed on the ballots seems to matter for convenience but not for security: if only a few random values are used, printing them on the ballot obviates the need for secure storage elsewhere.

5 Specific examples

5.1 California: multiple-winner first-past-the-post

Consider an election with multiple winners elected by first-past-the-post. The process here is identical to that which is currently performed for E2E-verifiability. Each assertion A_j can simply be the tally of candidate c_j.

For ballot b_i,

$$a_{ij} = \begin{cases} 1, & \text{if } b_i \text{ contains a vote for candidate } c_j, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

[12] This was suggested by Marc Rosen of Galois, Inc.

So commitment C_{ij} should be a commitment to 1 or 0, with a proof that the committed value is 1 or 0.

Then $A_j = \Sigma_i a_{ij}$, so assertion j can be checked by homomorphically summing the commitments and accepting the outcome if, for all $j = 1 \ldots, J$, the opened value of commitment C_j matches the announced tally A_j.
This check includes the proof of commitment range.

The null hypotheses $\mathcal{N}$ correspond to each case in which an announced loser's tally is higher than or equal to the winner's.

In the audit step, when a paper ballot has been retrieved, observers simply have to check whether a_{ij} has the right value as required above. If b_i is a vote for an announced loser but the commitment C_{ij} is not validly opened, the worst-case assumption is that a_{ij} is 0 if c_j is an announced loser, or 1 if c_j is an announced winner.

5.2 Instant runoff voting

Instant runoff voting (IRV) is used in numerous Commonwealth countries and some US state and local government elections. Each vote is a list of candidates in preference order. The social choice function first tests whether there is anyone with a strict majority of first-preference votes. If not, the candidate with the lowest tally is eliminated and their votes redistributed according to the next-listed preference on each ballot. This proceeds iteratively until one candidate has a strict majority.

To apply VAULT, the assertions $\mathcal{A}$ could be a description of each elimination in sequence, but a much more efficient audit could be conducted by using a set of assertions derived using the techniques of (Blom, Stuckey, and Teague 2019). In this case, $\mathcal{A}$ is a set of assertions about ballot preferences which, in conjunction, are sufficient to prove the election outcome (though not necessarily the exact elimination sequence that is claimed). For example, it would suffice to prove that one candidate received more first-preference votes than any other candidate received mentions (if it were true).

Using the notation of (Blom, Stuckey, and Teague 2019), define:

$\tilde{f}(c) =$ the number of first preference votes for c,
$\tilde{t}_S(c) =$ the tally of candidate c assuming the uneliminated candidates are those in set S

Note that $\tilde{f}(c)$ is the minimum tally c can possibly have, while $\tilde{t}_{\{c_1,c_2\}}(c_2)$ is the maximum tally that c_2 can possibly have in any election in which c_1 has not been eliminated. If $\tilde{f}(c_1) > \tilde{t}_{\{c_1,c_2\}}(c_2)$, then c_2 cannot possibly be eliminated before c_1.

The algorithm of (Blom, Stuckey, and Teague 2019) can produce various kinds of assertions that suffice, together, to prove that the reported winner truly won, and could therefore be immediately used for the set $\mathcal{A}$.

To take a simple example, suppose that in some particular IRV election with $n+1$ candidates, it happened to be the case that for all $j \neq n+1, \tilde{f}(c_{n+1}) >$

$\tilde{t}_{\{c_{n+1},c_j\}}(c_j)$. So define $a_j = \tilde{f}(c_{n+1}) - \tilde{t}_{\{c_{n+1},c_j\}}(c_j)$ for $j = 1\ldots,n$. Then c_{n+1} won the election if, for all $j = 1\ldots,n$, $a_j > 0$.

Although this is not always true, it turns out to be true surprisingly often in real IRV elections, in which case it provides a simple and efficient test of the announced election outcome.

The audit can proceed by testing the set of n assertions $\mathcal{A} \equiv \{a_j > 0\}_{j=1}^{n}$. More specifically, for ballot b_i,

$$a_{ij} = \begin{cases} 1, & \text{if } b_i \text{ has candidate } c_{n+1} \text{ as its first preference,} \\ -1, & \text{if } b_i \text{ has candidate } c_j \text{ preferred over } c_{n+1}, \\ 0 & \text{otherwise.} \end{cases}$$

So commitment C_{ij} should be a commitment to one of these values, with a proof that the committed value lies in the set $\{-1, 0, 1\}$.

Then $a_j = \Sigma_i a_{ij}$, so the public can check whether the CVRs satisfy the assertion $A_j \equiv \{a_j > 0\}$ by homomorphically summing the commitments and accepting the outcome if, for all $j = 1\ldots,J, a_j > 0$. This check includes the proof of proper range.

In the audit step, when a paper ballot has been retrieved, observers simply have to check whether a_{ij} has the right value as required above. If the commitment is not validly opened, the worst-case assumption is that a_{ij} is 1.

6 Overall risk-limit argument

Here we state our main security claim and sketch an argument to support it. The adversary controls the EA but not the verification algorithm. The security is based on the risk limit of a traditional RLA with plaintext CVRs—we assume correct functioning of the cryptographic aspects of that, including public verification that the random choices are correctly made and that the correct physical ballot is retrieved.

Recapping our setup:

- Let $\mathcal{A}$ be a set of assertions which, in conjunction, suffice to prove the accuracy of the announced election outcome.
- For each assertion $A_j \in \mathcal{A}$, S_j is the set of possible contributions to a_j for any valid ballot. (Note, it will usually be a range of integer values, but this is not necessary.)
- For each commitment C_{ij}, the authority proves and the verifier checks that C_{ij} is a commitment to a value in S_j.

We want to argue that the overall probability of mistakenly accepting a wrong election outcome (as defined by the physical ballots) is the (negligible) probability of breaking the cryptography, plus the risk limit of the RLA. We *don't* need to prove consistency across different commitments for the one ballot.

Claim. Let VAULT be parameterised with an RLA with Risk Limit α for plaintext CVR commitments. Then the risk limit obtained by substituting VAULT

for the traditional RLA procedure is at most $\alpha+\epsilon$, where ϵ is the combined probability of the attacker undermining the soundness property of either the ZKPs or the commitments, *i.e.*

- being able to open a commitment in two different ways, or
- producing a set-inclusion proof that passes verification for a value that is out of range.

Proof. (Sketch)

If the election outcome is wrong, then at least one of the null hypotheses is true. Wlog call it hypothesis N_1, and suppose it is negated by assertions A_1 and A_2. Then we have a series of commitments C_{i1}, C_{i2} for $i = 1, \ldots, n$ s.t. the homomorphically-added commitments

$$C_1 = \Pi_{i=1}^{n} C_{i1} \text{ and } C_2 = \Pi_{i=1}^{n} C_{i2}$$

can be opened as commitments to A_1 and A_2 (resp), and ZKPs ZKP_{i1}, ZKP_{i2} for $i = 1, \ldots, n$ s.t. z_{i1} (resp z_{i2}) passes verification for the statement that C_{i1} (resp. C_{i2}) commits to a value in S_1, (resp S_2) though in fact the claimed comparison between A_1 and A_2 is false according to the physical ballots.

If the authority can produce either a commitment opening to two different values, or a set-inclusion proof $ZKP_{S_i}(m_i)$ that passes verification though $m_i \notin S_1$, then cheating may succeed with probability greater than α.
We assume this happens with probability at most ϵ and, for the rest of this proof, assume that it has not happened.

Then the authority knows, for each C_{i1}, at most one tuple (m_{i1}, r_{i1}) that constitutes a valid opening (and likewise for C_{i2}). (Note that perfectly binding commitments, like El Gamal encryptions, get uniqueness automatically. *i.e.* there exists a unique valid opening, we don't have to assume that the authority knows only one.) Similarly, for the product commitments C_1 and C_2, the authority knows at most one valid opening (M_1, R_1) (resp (M_2, R_2)).

We have taken a random selection $\mathcal{I}$ of paper ballots (leaving aside for now the question of cheating on the predictability of those selections) and, for each of them, either had the corresponding commitment opened as (m_i, r_i) and checked whether it is a proper opening of C_{i1}, or had no commitment opened and made the worst-case assumption.

All commitments have been proven to come from some set, which has been checked to match the assumptions of the RLA. Some have been opened; others not, for which we made the worst-case assumption. Thus the process is equivalent to an RLA in which every ballot's contributing value was in S_1, with the CVRs being equivalent to the openable values for everything in $\mathcal{I}$, and the worst-case values for everything else.

So apart from the ϵ probability of cryptographic failure, everything about the audit is identical to an RLA (whichever RLA is being conducted) with m_i as the apparent/claimed CVR. Thus the overall probability of accepting a wrong election result is ϵ plus the risk limit of the RLA.

7 Privacy guarantees

VAULT exposes the exact contents of those ballots that are audited. This still allows for some coercion, because a randomly selected fraction of voters can prove that their ballots were part of the tally. There is also, always, the possibility of a full manual recount, which exposes all individual ballots. Hence the privacy guarantees of VAULT are usually better than an RLA that publishes plaintext CVRs, but they are not always strictly better and not better for all voters. However, if we consider an attacker who observes the WBB but not the full manual recount, then VAULT does not reveal extra information about those ballots that are not audited.

We assume that the election authority is trusted for privacy, and that the identity of the voter is separated from the CVR before it is committed on the WBB.

Claim. Against an attacker who observes the WBB but not the (possible) full manual recount, and assuming that the election authority is trusted for privacy, VAULT does not reveal information about votes that were not audited except what can be derived from the election outcome statement (Algorithm 1). The guarantee depends on the form of commitment: it is perfect if perfectly-hiding commitments are used, or computational if computationally-hiding commitments are used.

8 Conclusions

Risk-limiting audits are an important tool to ensure election integrity and to provide trustworthy public evidence that reported outcomes are correct: that tabulation errors did not result in reporting the wrong winner(s). However, the most efficient approach to RLAs—ballot-comparison audits—are publicly verifiable only if three conditions hold:

i. There is a public commitment to the full list of CVRs before the audit starts.
ii. The public can verify that applying the appropriate social choice function to the committed list of CVRs yields the reported election results.
iii. The public can verify how the contents of each CVR selected for audit compares to its corresponding paper ballots.

While these can be accomplished by publishing the entire list of CVRs as plain text, that would enable voter coercion. The only published approach to mitigate the risk of coercion while meeting (i)–(iii), SOBA [3], has never been used in a real election, possibly because of its complexity. We have shown that existing homomorphic tallying techniques used for end-to-end verifiability can make publicly verifiable, privacy-preserving ballot-comparison audits simpler, for instance, by publishing a complete list of homomorphically encrypted CVRs before the audit starts.

The minimal set of required cryptographic elements do not entail any change to voting systems, only post-processing the CVRs to create a set of cryptographic commitments for each ballot, and posting the results.

References

[1] Arapinis, M., V. Cortier, S. Kremer, and M. Ryan. 2013. "Practical Everlasting Privacy." In *International Conference on Principles of Security and Trust*, 21–40. Springer.

[2] Bañuelos, J.H., and P.B. Stark. 2012. "Limiting Risk by Turning Manifest Phantoms into Evil Zombies." arXiv.org. http://arxiv.org/abs/1207.3413.

[3] Benaloh, J., D. Jones, E. Lazarus, M. Lindeman, and P.B. Stark. 2011. "SOBA: Secrecy-Preserving Observable Ballot-Level Audits." In *Proceedings of the 2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE '11)*. USENIX. http://statistics.berkeley.edu/~stark/Preprints/soba11.pdf.

[4] Blom, M., P.J. Stuckey, and V. Teague. 2019. "Risk-Limiting Audits for IRV Elections." *arXiv Preprint arXiv:1903.08804*.

[5] Boneh, D., and V. Shoup. 2016. "A Graduate Course in Applied Cryptography." *Draft of a Book, Version 0.4*.

[6] Bünz, B., J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. 2017. "Bulletproofs: Efficient Range Proofs for Confidential Transactions." *IEEE SP*.

[7] Camenisch, J., R. Chaabouni, and A. Shelat. 2008. "Efficient Protocols for Set Membership and Range Proofs." In *International Conference on the Theory and Application of Cryptology and Information Security*, 234–252. Springer.

[8] Cramer, R., I. Damgård, and B. Schoenmakers. 1994. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols." In *Annual International Cryptology Conference*, 174–187. Springer.

[9] Demirel, D., J. Van De Graaf, and R. Araújo. 2012. "Improving Helios with Everlasting Privacy Towards the Public." *EVT/WOTE* 12.

[10] Katz, J., and Y. Lindell. 2014. *Introduction to Modern Cryptography*. CRC Press.

[11] Lewis, S. J., O. Pereira, and V. Teague. 2019. "Ceci N'est Pas Une Preuve: The Use of Trapdoor Commitments in Bayer-Groth Proofs and the Implications for the Verifiabilty of the Scytl-SwissPost Internet Voting System."

[12] Lindeman, M., and P.B. Stark. 2012. "A Gentle Introduction to Risk-Limiting Audits." *IEEE Security and Privacy* 10: 42–49.

[13] Mao, W. 1998. "Guaranteed Correct Sharing of Integer Factorization with Off-Line Shareholders." In *International Workshop on Public Key Cryptography*, 60–71. Springer.

[14] Moran, T., and M. Naor. 2006. "Receipt-Free Universally-Verifiable Voting with Everlasting Privacy." In *Annual International Cryptology Conference*, 373–92. Springer.

[15] Moran, T., and M. Naor. 2010. "Split-Ballot Voting: Everlasting Privacy with Distributed Trust." *ACM Transactions on Information and System Security (TISSEC)* 13 (2). ACM: 16.

[16] National Academies of Sciences, Engineering, and Medicine. 2018. *Securing the Vote: Protecting American Democracy*. Washington, DC: The National Academies Press. https://doi.org/10.17226/25120.

[17] Ottoboni, K. 2019. "Classical Nonparametric Hypothesis Tests with Applications in Social Good." PhD thesis, University of California, Berkeley.

[18] Pedersen, T.P. 1991. "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing." In *Annual International Cryptology Conference*, 129–140. Springer.

[19] Stark, P.B. 2008. "Conservative Statistical Post-Election Audits." *Ann. Appl. Stat.* 2: 550–581.

[20] Stark, P.B. 2009. Risk-limiting post-election audits: P-values from common probability inequalities, *IEEE Transactions on Information Forensics and Security*, 4: 1005–1014.