Risk-limiting Audits: A Practical Systematization of Knowledge

Matthew Bernhard

VotingWorks

Abstract. Risk-limiting audits (RLAs) are broadly accepted as the gold standard for tabulation audits: when I count ballots using software, an RLA provides software-independent evidence that tabulation declared the correct winner. While there have been many advances in RLAs over the last 14 years, many of the underlying assumptions and practical applications of RLAs have gone unexamined. In this paper, I present a review of existing RLA techniques, providing a concise definition of an RLA, examining its underlying assumptions, and discussing how RLAs work in practice, all from the perspective of the maintainer of a popular RLA tool. I present several attacks which can cause RLAs to fail to provide evidence of the correctness of an election outcome. Finally, I provide discussion on the RLA's place in the landscape of election security and observations about the value proposition RLAs present.

1 Introduction

Post-election audits have long been a tool to evaluate election processes. They serve as an opportunity for election officials to check their work and provide transparency into election processes for voters. Additionally, post-election audits have long been a point of emphasis for election security, as they often provide robust means of achieving important properties like software independence [38]. In the wake of the 2016 presidential election in the United States, post-election audits, and risk-limiting audits in particular, have become widely adopted as a means of securing and improving the election process [13, 20, 27, 31, 35].

Much work has been conducted on post-election audits as a means of building evidence-based elections [52]. Stark produced the concept of a risk-limiting audit (RLA), a post-election tabulation audit which can provide a desired level of confidence in the election outcome with an amount of work that depends on the desired level of confidence [46].

Interwoven in much of the work around post-election audits are subtle assumptions and an implicit threat model. Most early efforts focused largely on the threat model of the election itself, with audits as a defense. While efficiency is a common theme, it is usually framed as pragmatism, not a feature of the audit to be safeguarded. Stark and Wagner contemplate ways in which the audit can be compromised, if, e.g., the paper trail is not properly maintained [52]; however they do not formally consider how a compliance audit can factor into the critical functions of an RLA.

In this paper I explore the assumptions underlying risk-limiting audits, discuss how they are conducted in practice, and what these things mean for the threat model for RLAs. I start by defining what an RLA is and its capabilities in the next section. I then examine how RLAs are performed in practice in Section 3. In Section 4 I describe attacks against RLAs that defeat their primary functions before concluding in Section 5.

2 What is a risk-limiting audit?

Broadly speaking, the goal of an RLA is to provide some degree of confidence in the outcome of an election while doing some amount of work. The trade-off is centrally between the level of confidence (framed as the "risk limit") and the amount of work required to reach it (usually in terms of the number of ballots that need to be counted by hand). However, there are numerous assumptions baked into RLAs that are often not explicitly stated, though works like [8, 19, 52] have attempted to make these assumptions more concrete.

The core of an RLA lies with the generation and preservation of evidence for an election outcome [19, 28, 52]. If an election outcome is produced by the automatic tabulation of ballots, the tabulators may be compromised in such a way that is not obvious (e.g. they may flip a few votes in a close race to change the outcome). Absent efforts to check the tallies, this process does not provide sufficiently verifiable evidence that the election's outcome is correct. This is the problem that RLAs attempt to address.

RLAs assume that there exists a "true" election outcome. Ideally the true election outcome would be produced by a social choice function based on the preferences of the voters. For instance, if first-past-the-post is the social choice function, and most voters prefer Alice over Bob, then the "true" outcome is that Alice wins. Unfortunately, political preferences are often inconsistent even within individuals [3], so elections must force people to commit to their preferences. This can be done in a variety of ways, but the most common form is via a secret ballot onto which voters imprint their preferences, typically by selecting from a provided list of choices (though occasionally something more complex). Since the voter's choices on the ballot are assumed to align with their preferences, the ballot is the record of the voter's intent, considered the "true" expression of each voter's preferences. While some voting systems allow voters to mark ballots in a purely mechanical or electronic way, as on a direct-recording electronic voting machine (DRE), these methods do not as of yet produce a durable record of the vote that can be independently verified [10] (i.e. they aren't software-independent [38]). Therefore, RLAs rely on *paper* ballots as the source of true preferences for an election outcome.

Assumption 1. Paper ballots reflect voters' true preferences.

This assumption means that RLAs do not attempt to characterize mistakes voters may make when marking their ballots, even though mistakes may be prevalent enough to change the outcome [12, 41]. Nor do they account for any other mechanism that can cause the paper record of the vote to not reflect the voters' preferences (or the choices voters make to reflect those preferences), like malicious software on ballot marking devices [1, 11, 24].

Another assumption that RLAs make is that the set of ballots being examined is complete. If the ballots examined in the RLA do not comprise of all of the valid ballots used to tabulate the election results, then an RLA's output is largely meaningless. This also means that RLAs require stringent chain-of-custody records to provide verifiable evidence that the paper trail is well maintained [52].[1]

Assumption 2. The paper trail examined by the RLA is complete and correct.

To make an election process evidence-based and independently verifiable, RLAs rely on a means of tabulation that does not rely on software: hand counting. While research has found that humans do make mistakes when counting ballots [21], RLAs implicitly trust that hand counts are accurate. This assumption may be reasonable, as audits have a much greater degree of transparency and public visibility than software running on scanners, but has largely been unexamined in the literature. In summary, RLAs treat the result of hand counts as ground truth for the "real" election outcome. Since hand counts do not rely on software (or do not rely on software in a way that violates software independence; see Section 3), they are a software-independent mechanism for evidence evaluation. In other words, RLAs assume that a hand count of all the ballots in an election always produces the correct outcome [46].

Assumption 3. Full hand counts always produce "true" election outcomes.

Even though we now have a definition of what a true evidence-based election process would look like, RLAs rely on one more key assumption. If hand counts were truly the best method of conducting elections, then surely they would be the most widely adopted means of tabulating votes. However, this is obviously not the case in countries all over the world [2, 42, 57]. Therefore there must be benefits to software tabulation that outweigh its costs.[2] RLAs thus seeks to gain confidence that an election outcome is correct while not doing a full hand count, if possible.[3]

Assumption 4. Hand counting is undesirable and should be minimized.

We now have a clearly defined problem statement for RLAs:

Definition 1. Risk-limiting audits are an evidence-based process to provide a specified level of confidence in an election outcome while minimizing hand counting.

[1] This is not a given, as Enguehard et al. have documented, among others [17].

[2] Namely, cost, accuracy, speed, and repeatability of tabulation, though see, e.g., [53].

[3] Though RLAs may also be performed on hand-tabulated elections as well [40].

3 Logistics of RLAs

Now that I have established what an RLA is attempting to accomplish, I can set about describing how it does so. There are several dimensions to performing an RLA that each introduce unique challenges to the threat model.

3.1 Hypothesis testing

RLAs draw a sample of ballots (typically uniformly at random [29] though occasionally not [5, 46]), hand count the sample, and then use the data produced by the hand count to evaluate the election result produced by software tabulation. There are numerous details here that I will discuss later on in this section, but this description suffices for now.

Given software-produced tabulation and our hand count data, an RLA seeks to assert or reject the hypothesis that the election outcome is correct. Specifically, RLAs usually take the software-produced outcome as an alternative hypothesis (though other alternatives are possible; see [23, 58]), and that the outcome is wrong as the null hypothesis. In this case, "the outcome is wrong" is formalized as the true outcome either preferring a candidate that did not win or a tie. These two hypotheses are mutually exclusive (though it may take some massaging depending on the social choice function) and exhaustive (by definition), so if the null hypothesis is rejected, then one can conclude to a degree of confidence that the alternative is true.

As this is a hypothesis test, the test can have four outputs depending on the underlying "true" hypothesis and the efficacy of the test. A **true positive** results when the null hypothesis is correctly rejected, i.e. the reported outcome is correct and the audit found evidence to support that to the specified risk limit. A **true negative** results when the reported election outcome disagrees with the outcome that would be produced by a full hand-tally. In this case, RLAs are designed to escalate to a full hand-tally before concluding that the reported election result is wrong. This feature makes a **false positive** effectively impossible: a correct reported outcome can never be overturned by an RLA [46], as outcomes can only be overturned by a full recount, at which point the true outcome will be known. A **false negative** results when the election outcome *should* be overturned, but the audit incorrectly terminates after fewer than all of the ballots have been audited. False negatives can occur by bad luck: if the sample data just happens to strongly favor the reported winner, the audit may conclude that the reported winner really did win even if that is not the case. The largest chance of this occurring is the **risk limit**.

The output of a hypothesis test is a **p-value** (occasionally called the risk measurement), an estimation of how likely the data would have occurred by random chance. If this value is below the risk-limit, the audit stops. If not, it draws more ballots and generates another p-value until all of the ballots have been counted or the p-value is below the risk-limit.

3.2 Evaluating hypotheses

In order to test the hypothesis that the reported election outcome is correct, RLAs require a means of evaluating evidence. There are two primary ways paper ballots are used to evaluate the reported outcome: **polling** audits and **comparison** audits [29]. There are also two **units** that audits can operate over: individual ballots or batches of ballots.

In a polling audit, a sample of ballots is drawn, and the social choice function is computed over the sample. The resulting tally is then used to evaluate the null and alternative hypotheses [26, 33, 51, 58]. Polling audits typically require dramatically larger sample sizes than comparison audits, as the evaluation being done tends to be less sensitive. However, polling audits require significantly less infrastructure to perform than comparison audits [29, 58].

Comparison audits, rather than relying on the social choice function to evaluate hypotheses, directly evaluates the evidence generated by the election. Each ballot (or batch) is audited, and the result of the audit is compared to the record of how that ballot (or batch) was tabulated. If discrepancies are found between the audited ballot and the reported tabulation, this indicates that an outcome-changing event may have occurred [22, 46, 48–51].

Finally, there are also *hybrid* auditing methods, most notably SUITE [34] and SHANGRLA [51]. These audits break the set of ballots into separate *strata*, where different types of evaluation (polling and comparison), different units of auditing (ballot or batch), or both, are used in each stratum.

3.3 Collecting evidence

In order to collect evidence to support or refute a hypothesis, RLAs must draw a sample from a population of ballots. We may not trust a reported election result, but at a minimum we must know how many paper ballots there are (i.e. how large the population is) and where they are (i.e. how can a specific ballot be found) in a software independent way.

The first piece of data, the population size, can be determined several ways, for example examining the voter check-in data to see how many voters checked in to vote (though this data must be verified against the chain of custody of paper ballots). The latter requires a bit more work depending on the storage procedures for the ballots. We can imagine a warehouse containing all of the ballots in a contest, where ballots are stored in boxes. In order to audit ballot n, then, we need to know which box it is contained in. This data is called a **ballot manifest**, and includes a software-independent accounting of ballot storage [29, 52]. As we shall see in Section 4, errors in the ballot manifest can cause the RLA to fail to achieve its goals.

Now that we have established what our population is, we can sample from it. There are a variety of sampling techniques available. The simplest and most common is sampling uniformly at random after the results have been fully tabulated and reported. However, Ottoboni et al. examined sampling uniformly before the results have been fully tabulated [33]. Sampling especially in batch audits may be done with respect to the amount of possible "error" that can be found in a unit [5,47].

To generate a sample, the manifest is used to create a "master list" of ballots (or batches), where each ballot gets a sequential number (e.g. if the last ballot in the first box is ballot n, then the first ballot in the next box is $n + 1$ and so on). In principle, ballots can be sampled at random by merely rolling enough dice to cover the range of ballot indices, modulo the population size [15]. In practice, this is far too laborious, so pseudo-random number generating programs (PRNGs) are often used to generate a list of ballots to sample [15,33].

Once a ballot is sampled, the manifest is used to figure out which container that ballot is stored in, and a variety of methods can be used to find the specific ballot in question (counting down in the stack of ballots or using a counting scale, for example [36]). Sridhar et al. proposed an alternative method for sampling an individual ballot from a batch, called k-cut: cutting the stack of ballots like a deck of cards. They demonstrated that cutting the stack of ballots six times in randomly generated places was sufficient to get a uniformly sampled ballot [43].[4] Batch-level audits do not require these techniques, as once a batch is sampled all of the ballots it contains are part of the sample and are hand counted.

3.4 Examining the evidence

Once the ballots to be examined have been drawn, RLAs require a means of evaluating the evidence. The nature of this evaluation depends heavily on the specific hypothesis being tested. BRAVO ballot polling audits examine hypotheses about how the margin is distributed *proportionally* to each candidate [26,33,34,58], while the type of ballot polling described in [34,51] looks at the margin in terms of integer votes. In both cases, ballot polling audits test the hypothesis that the winner actually won by examining the margin in the sample against the null hypothesis (the winner didn't win; in most cases the margin under the null is 0, i.e. a tie) and an alternative hypothesis (typically the reported margin, but occasionally not [23]). Most extant ballot polling audits examine the likelihood ratios of the null and the alternative given the tally of a sample [26,29,32,37,55,58], though methods like [39,51] use other test statistics. In any case, these audits require very little information to evaluate their hypotheses: the null margin, alternative margin, and a hand tally of the sampled ballots. If their test-statistic outputs a p-value below the risk limit, the audit stops. Otherwise, more ballots are sampled.

Comparison style audits also examine the margin in votes, but rather than examine a tally of the votes in the sample, comparison audits directly compare the sampled units against their reported counterparts. If a ballot's audited record differs from its reported record, this is deemed an error. Errors can work in multiple

[4] K-cut may not be suitable for ballot comparison audits, since they need the ability to look up a sampled ballot's cast vote record (see below). If the ballots do not have an identifier (imprinted on the ballot at the time of scanning), the ballots must be kept in the order they were scanned.

ways: **overstatements** are errors which cause the margin to be overstated, i.e. the winner didn't win by as much as was reported. **Understatements** are errors which indicate that the winner actually won by more than originally reported. If the number of overstatements is larger than the margin of victory, then the reported outcome is wrong.

In a ballot comparison audit, if a ballot is sampled, it is examined and the votes it contains are directly compared to how it was originally tabulated [46, 50, 51]. These audits need an additional input: a record of how each individual ballot was tabulated (called a **cast vote record** or CVR) that can be compared against an examination of the corresponding paper ballot.

For batch comparison audits, the hand-tallied batch totals are compared to the tabulator results for each batch. Rather than a CVR, batch comparison audits require a batch totals file that contains how each batch was tabulated by software, which is then compared to hand tallies. It should be noted that ballot comparison audits are just special cases of batch comparison audits where each batch contains only one ballot [22, 48].

3.5 Tools for performing an RLA

Much of the logistical support and calculation in RLAs can in principle be done without reliance on any software. However, this is practically infeasible in most contexts (e.g. if an RLA needs to draw 5,000 ballots, rolling enough dice to draw a sample would take an inordinate amount of time). Moreover, election officials (or other auditors) may not have the expertise in statistics required to perform the calculations necessary to evaluate evidence. Finally, coordinating an audit in a large jurisdiction, like a state, becomes intractable when dozens or hundreds of ballot manifests, CVRs, and other election information must be collated.

To address these problems, numerous software tools have been developed to support the logistics of an audit. In order to preserve software-independence, the inputs and outputs of the software need to be made available. Publishing this data enables anyone to independently verify that the software conducted the audit correctly (either by working it through by hand or by using different, trusted software). To my knowledge, all existing RLA software is open source, which aids in the verifiability of the software's correctness and also supports independent evaluation of audits.

Tools for performing ballot polling audits have been made available by Stark [44] and VotingWorks [56]. Additional ballot polling software has been made available by Ottoboni et al. [33], Stark [51], and Zagorski et al. [58]. Tools for performing comparison audits have been made available by Stark [45, 51], VotingWorks [56], and Free and Fair [18]. McBurnett also has a tool for older batch comparison audits including SAFE [30], NEGEXP [5], and PPEB [47]. Finally, tools for performing hybrid audits have been provided by Ottoboni et al. [34] and VotingWorks [56].

As we shall see in the next section, these tools encompass a variety of communications channels between participants in the audit, and can present risk.

3.6 How does an RLA actually work?

We have covered much of the inputs and outputs of RLAs so far, but I have not described the process in full nor identified the stakeholders.

Stakeholders RLAs are fundamentally transactions between evidence havers and evidence seekers. Election officials are in charge of the software that does the tabulation as well as the paper trail. For large audits, there are often two types of election official: **audit administrators**, in charge of coordinating the data collection and entry procedures, and **local election officials** who have custody of the paper trail and do the work of pulling and examining ballots. The election officials rely on **audit software** to conduct the audit. The **public** observes the audit and seeks to be convinced that the paper ballots match the reported election outcome. Finally, **attackers** seek to disrupt this process.

Process The RLA process is sequential, and is described as follows (and depicted in Figure 1):

1. The election ends, a final tally is produced, and an outcome is declared.
2. Audit administrators initialize the audit with information about the contest to be audited (the contest name, number of winners, candidates, tally of votes for each candidates, and outcome), as well as a list of authorized participants if the audit is happening in multiple places and involves distributed access to the tool (local election officials who possess the paper ballots).
3. Local election officials collect and submit all necessary inputs, including a ballot manifest and, if performing a comparison audit, batch totals or CVRs.
4. A ceremony for initializing a PRNG is conducted, where some physical randomness (e.g. dice) is used to seed the PRNG.
5. Based on the contest information (the margin of victory) and previous samples (if not in the first round of the audit), a sample size is estimated (i.e. the number of ballots or batches to audit is determined).
6. The ballots to be sampled are identified by the PRNG, and information about which ballots or batches are to be examined (called a retrieval list) is distributed to local election officials.
7. Local election officials retrieve the ballots in their jurisdiction, examine them, and submit the information to the audit administrators (either out of band or through the audit software).
8. Once all information about the sample is entered, the audit administrator computes a p-value for sample based on the evidence. If the p-value is greater than the risk limit, go back to step 5, generate a new sample and collect more evidence. If all ballots have been sampled, stop and announce the outcome of the sampled data as the new outcome. If the p-value is less than or equal to the risk limit, halt the audit and announce that the outcome is confirmed.
9. At the conclusion of the audit, all information about the audit is published, including the manifests, CVRs, the random seed and PRNG algorithm, and adjudication of the sampled ballots so that anyone may verify the results.

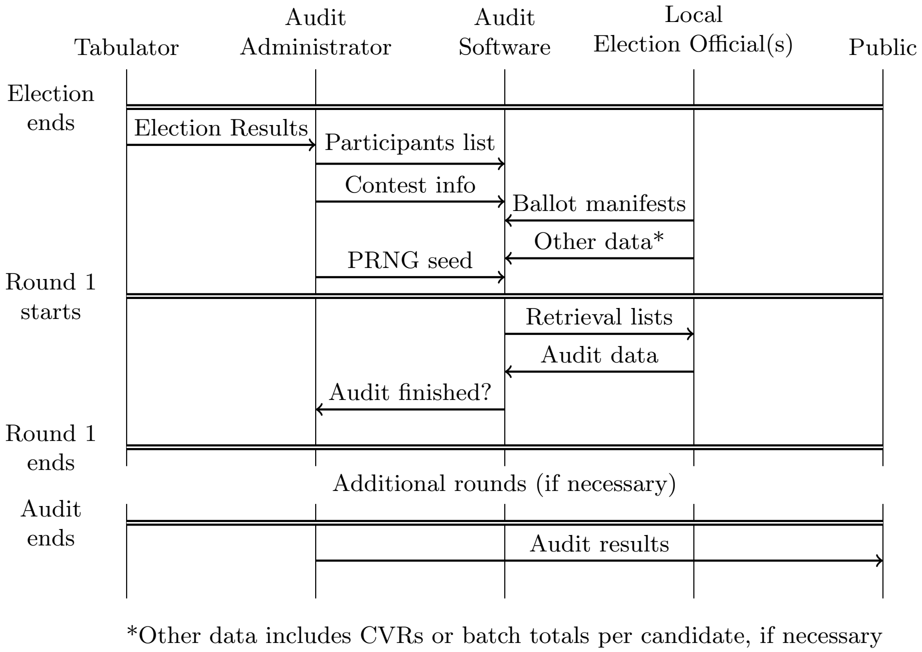

Fig. 1. A process diagram of an RLA—Each column represents a stakeholder. Arrows between stakeholders indicate communication of some sort between two stakeholders. At the start of each round, the audit tool computes a sample size and draws a sample of ballots, which are distributed to local election officials. Having received their retrieval lists, LEOs sample ballots as indicated by the retrieval list and enter the audit data into the audit software. After all ballots in round have been entered, the software computes a p-value for the audit so far, and notifies the audit administrator whether the audit has met the risk limit, or if more ballots need to be sampled. Note that the depicted protocol is an insecure strawman, and the communication channels must be secured and the data must be made public for the audit to be successfully verifiable (see Section 4.5).

4 Attacks and Defenses

An overview of the audit process and relevant communication channels can be seen in Figure 1. As we shall see shortly, the way the process is laid out presents several opportunities for an attacker to compromise the audit. Revisiting our assumptions about RLAs from Section 2, we can define a major and a minor goal for our attacker:

- **Major goal:** Cause an RLA to stop early even though a full hand count would indicate a different result more frequently than the risk limit allows.
- **Minor goal:** Force the RLA to a full hand count.

Recall from Section 3.1 that the hypothesis testing at the core of an RLA can indicate four distinct outcomes. The "true" outcomes are where an RLA does what it is supposed to: either correctly stop after seeing sufficient evidence or correctly escalating to a full recount and overturning an election. An attacker seeks to cause a "false" outcome: either incorrectly concluding that the reported outcome was correct or overturning a correct result. As we discussed earlier, the latter outcome (a false positive) is impossible if the paper trail is preserved, so an attacker can at best cause a false negative (accepting an incorrect result).

This last point is our attacker's major goal: cause a false negative with probability greater than the risk limit. This might occur if the attacker had also compromised the tabulator and wished to remain undetected.

Additionally, recall that RLAs also assume that hand counting all of the ballots are undesirable. Therefore a minor goal for an attacker is to cause a full recount every time an RLA is performed (intended to waste resources).

In general, an attacker who can accomplish the major goal can also accomplish the minor goal, but not the other way around. Additional goals like sowing public distrust are considered out of scope for now. With these goals in mind, we present several attack scenarios.

4.1 An active network attacker

In this scenario, the attacker is able to intercept, alter, and transmit messages to any stakeholders. This attacker cannot alter the paper trail, audit tool, or PRNG generation ceremony. However, they can impersonate all parties and use the audit tool to generate expected output.

Forged sample sizes This attacker can forge the expected sample sizes and cause them to be larger or smaller than required. This achieves the minor goal, as they could indicate that the audit needs to examine all of the ballots.

Forged retrieval lists This attacker can issue forged retrieval lists directly to local elections officials (LEOs). In doing so, they can bias the sample by selecting ballots from jurisdictions that lean more towards the declared winner. If the audit samples these ballots, it can falsely conclude that the declared result was correct.

Forged audit results This attacker can similarly forge messages from the LEOs to the audit tool and input fake audit data that causes the software to convince that the audit can terminate.

Forged messages from the audit administrator This attacker can announce audit results to the public early or otherwise in a way that sows doubt about the correctness of the election process.

This attacker's power largely comes from the lack of out-of-band communication between stakeholders. To mitigate this attack, an authenticated channel should be established between all parties to ensure that only authenticated parties are submitting and generating audit data. Standard authentication techniques, like TLS, and the use of restricted domains and authenticated email accounts [54] apply here. Furthermore, independent verification of the audit would catch many of these attacks, as the sample sizes, retrieval lists, and results are deterministic and can be checked.

4.2 Compromised audit software

An attacker can still gain full control over the audit if they compromise the audit software, even without actively attacking the RLA's communication channels. All of the forgery possibilities above apply, except announcing the result. However, presuming an honest audit administrator, if the tool provides an incorrect audit result the audit administrator may just announce the audit result provided by the tool without verification.

In order to defend against a compromised audit tool, software-independent bookkeeping is critical. The random seed, PRNG algorithm, ballot manifests, and audit results must be published through a channel that doesn't rely on the tabulator or the audit software (e.g. by scans of paper documents uploaded to the audit administrator's website) [19,25,52]. With this data (and CVRs or batch totals files), in principle anyone in the public can verify the audit results by hand or by using different, trusted software.

4.3 Compromised audit administrator

In this case, the audit administrator is compromised, either because they themselves wish to confirm an incorrect outcome or because malware resident on the device they're using to interact with the audit software does. The defenses from Section 4.2 protect against this except for one attack on comparison audits.

If the audit administrator announces wrong election results and tampers with the data provided to the auditing software, it is possible for the audit to reject the null hypothesis when it shouldn't. By entering incorrect contest totals but providing the correct tabulation data needed for comparison audits, the audit will confirm the outcome even though the paper ballots, if counted by hand, would result in a different outcome. This is because comparison audits assume that the contest data is correctly entered, and that any discrepancies between the audited ballots and the tabulation data will catch incorrect contest data. By declaring the wrong result and entering the wrong vote totals, but using uncompromised CVRs or batch totals files, the comparison audit will terminate without seeing discrepancies, "confirming" that the announced outcome indeed corresponds to the paper ballots. However, this will not be the case.

In order to prevent this attack, the contest data and the CVRs or batch totals must be made public so that the contest data can be verified against the CVRs or batch totals. However, in ballot comparison audits, published CVRs can enable coercion (the so-called Sicillian attack, for example [10]). For this reason, states like Colorado explicitly forbid publishing CVRs in plain text [14]. Therefore, CVR values need to be hidden (except for audited ballots). To achieve this, methods like [8] split up CVRs per-contest and produced commitments, while [9] homomorphically encrypts CVRs and publishes the encrypted CVRs and a decryption key for the CVR totals before the audit starts. Anyone can then sum the encrypted CVRs and decrypt the result to verify that the contest data entered by the audit administrator matches the files to be compared. The ballots which are audited also have their plaintext CVR published, which allows the public to verify that the audit performed as expected.[5]

4.4 Compromised local election official

A compromised local election official represents the greatest threat to an RLA. LEOs are the custodians of the paper trail, so if they desire to subvert the audit, they can do so trivially by tampering with paper ballots. However, RLAs assume that sufficient compliance auditing [52] is performed (see Section 2). Still, LEOs (or malware on their behalf, below LEO refers to both cases) may subvert the audit in a few other ways.

Forged manifests If an LEO or collection of LEOs misrepresents how many ballots they have in their custody, they can influence the result of the RLA. For instance, an LEO in a jurisdiction that votes more for the winner can claim more ballots than they actually have. In a comparison audit, this would cause the audit to recount all of the ballots, as ballots claimed to exist but not found are counted as the worst kind of error [6]. In a ballot polling audit using k-cut, things are even worse, as unless the number of ballots is counted ahead of time and found to be too small, samples can be drawn without noticing anything is wrong. This is why it is critical for LEOs to produce software independent ballot manifests that can be published out of band and independently verified.

Forged audit data LEOs may also enter incorrect data about the audited ballots (as in the network attacker above). This can cause the audit to finish early. Ideally, public observation would prevent this, and paper records about the audit can be made available for verification.

4.5 Summary of defenses

I have detailed a wide variety of attacks, and mentioned some defenses in passing:

- Authenticated communication channels between all parties, including authenticated access to the audit tool
- Publication of all audit data through an authenticated channel, like the audit administrator's website
- Commitment to audit data before use (e.g. the contest information should be known before the audit starts and verifiably not changed afterwards)
- Use of CVR encryption, where applicable [9]
- Software independent record keeping of audit data [19]
- Public observation of all audit processes
- Compliance audits that enforce chain of custody for the paper trail [52]
- External verification of all audit inputs and outputs using a trusted audit tool

[5] An implementation of [9] can be found here: [4].

5 Conclusion

In this paper I have discussed the theoretical and practical facets of risk-limiting audits. I have examined the underlying assumptions made by RLAs and identified how those play out in practice. I have shown that numerous attacks against RLAs exist which can cause an RLA to fail to accomplish its primary goals, and also presented several defenses which mitigate these attacks.

Despite my analysis, I have largely deferred a critical consideration. RLAs exist to provide trust in election outcomes. They rely on a durable evidence trail, large-scale transparency processes, and at times complex statistics to provide proof that a reported outcome really does match all available evidence. However, it might be worth considering whether they succeed. Absent any attack, even if a risk-limiting audit is carried out faithfully and correctly, does it make voters more confident in election outcomes? Future research is needed on this point, as after the 2020 election, more U.S. states performed risk-limiting audits than ever before, all of which found significant evidence that the election outcome was correct. Yet, a significant fraction of the population still maintains that the reported outcome was wrong [7]. If RLAs don't provide confidence, is there value in them at all?

Recent events in the U.S. state of Georgia may provide some clarity on the issue. Georgia is a historically Republican state, with all executive offices held by Republicans, including the chief elections officer Brad Raffensperger. In an environment where much of the Republican party refused to accept the election result, going so far as to vote against certifying it in the U.S. federal legislative bodies, the Republican executives in Georgia stood by their election results, in which a Democratic presidential candidate won the state of Georgia for the first time in 28 years. Secretary Raffensperger repeatedly cited Georgia's risk-limiting audit as a major reason for his confidence [20]. Even if RLAs may not be intelligible or convincing to the public at large, there is significant value in the fact that they provide election officials (and interested members of the public) confidence in the election results. In a time where misinformation is wreaking havoc the world over [16], RLAs bolster the bastion of democracy.

Acknowledgments Thanks to Mark Lindeman, Monica Childers, Ben Adida, Kevin Skoglund, Dan Wallach and the anonymous reviewers for their insightful feedback.

References

1. Appel, A.W., DeMillo, R.A., Stark, P.B.: Ballot-marking devices cannot ensure the will of the voters. Election Law Journal: Rules, Politics, and Policy **19**(3), 432–450 (2020)
2. Aranha, D.F., van de Graaf, J.: The good, the bad, and the ugly: two decades of e-voting in Brazil. IEEE Security & Privacy **16**(6), 22–30 (2018)
3. Arcuri, L., Castelli, L., Galdi, S., Zogmaister, C., Amadori, A.: Predicting the vote: Implicit attitudes as predictors of the future behavior of decided and undecided voters. Political Psychology **29**(3), 369–387 (2008)

4. Arlo CVR Encryption. https://github.com/votingworks/arlo-cvr-encryption
5. Aslam, J., Popa, R., Rivest, R.: On Auditing Elections When Precincts Have Different Sizes. In: 2008 USENIX/ACCURATE Electronic Voting Technology Workshop, San Jose, CA, 28–29 July (2008), http://www.usenix.org/event/evt08/tech/full_papers/aslam/aslam.pdf, retrieved 30 May 2011
6. Banuelos, J.H., Stark, P.B.: Limiting risk by turning manifest phantoms into evil zombies. arXiv preprint arXiv:1207.3413 (2012)
7. Barry, D., McIntire, M., Rosenberg, M.: 'Our President Wants Us Here': The Mob That Stormed the Capitol. New York Times (2021), accessed from https://www.nytimes.com/2021/01/09/us/capitol-rioters.html
8. Benaloh, J., Jones, D., Lazarus, E., Lindeman, M., Stark, P.: SOBA: Secrecy-preserving Observable Ballot-level Audits. In: Proceedings of the 2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE '11). USENIX (2011), http://statistics.berkeley.edu/~stark/Preprints/soba11.pdf
9. Benaloh, J., Stark, P.B., Teague, V.: VAULT: Verifiable Audits Using Limited Transparency. E-Vote-ID 2019 p. 69 (2019)
10. Bernhard, M., Benaloh, J., Halderman, J.A., Rivest, R.L., Ryan, P.Y., Stark, P.B., Teague, V., Vora, P.L., Wallach, D.S.: Public evidence from secret ballots. In: International Joint Conference on Electronic Voting. pp. 84–109. Springer (2017)
11. Bernhard, M., McDonald, A., Meng, H., Hwa, J., Bajaj, N., Chang, K., Halderman, J.A.: Can Voters Detect Malicious Manipulation of Ballot Marking Devices? In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 679–694. IEEE (2020)
12. Byrne, M.D., Greene, K.K., Everett, S.P.: Usability of voting systems: Baseline data for paper, punch cards, and lever machines. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 171–180 (2007)
13. Colorado Secretary of State: Colorado Secretary of State Jena Griswold Certifies the State's 2020 General Election. https://www.sos.state.co.us/pubs/newsRoom/pressReleases/2020/PR20201208CertifyElectionResults.html
14. Colorado Secretary of State: Rule 25. Post-election audit. In: Election Rules, chap. 25.2.4. Colorado Secretary of State (2021), accessed from https://www.sos.state.co.us/pubs/rule_making/CurrentRules/8CCR1505-1/Rule25.pdf
15. Cordero, A., Wagner, D., Dill, D.: The role of dice in election audits—extended abstract. In: IAVoSS Workshop on Trustworthy Elections. Citeseer (2006)
16. Del Vicario, M., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., Stanley, H.E., Quattrociocchi, W.: The spreading of misinformation online. Proceedings of the National Academy of Sciences **113**(3), 554–559 (2016)
17. Enguehard, C., Graton, J.D.: Machines à voter et élections politiques en France: étude quantitative de la précision des bureaux de vote. Cahiers Droit, Sciences & Technologies (4), 159–198 (2014), original in French. Translated via Google.
18. Free, Fair: ColoradoRLA. https://github.com/FreeAndFair/ColoradoRLA
19. Garland, L., Lindeman, M., McBurnett, N., Morrell, J., Schneider, M.K., Singer, S.: Principles and best practices for post-election audits. https://verifiedvoting.org/publication/principles-and-best-practices-for-post-election-tabulation-audits/ (2018)
20. Georgia Secretary of State: Historic First Statewide Audit of Paper Ballots Upholds Result of Presidential Race. https://sos.ga.gov/index.php/elections/historic_first_statewide_audit_of_paper_ballots_upholds_result_of_presidential_race
21. Goggin, S.N., Byrne, M.D., Gilbert, J.E.: Post-election auditing: effects of procedure and ballot type on manual counting accuracy, efficiency, and auditor satisfaction and confidence. Election Law Journal: Rules, Politics, and Policy **11**(1), 36–51 (2012)

22. Hall, J.L., Miratrix, L.W., Stark, P.B., Briones, M., Ginnold, E., Oakley, F., Peaden, M., Pellerin, G., Stanionis, T., Webber, T.: Implementing risk-limiting post-election audits in California. In: EVT/WOTE 2009. USENIX Association (2009)
23. Huang, Z., Rivest, R.L., Stark, P.B., Teague, V.J., Vukcevic, D.: A unified evaluation of two-candidate ballot-polling election auditing methods. In: International Joint Conference on Electronic Voting. pp. 112–128. Springer (2020)
24. Kortum, P., Byrne, M.D., Whitmore, J.: Voter Verification of BMD Ballots Is a Two-Part Question: Can They? Mostly, They Can. Do They? Mostly, They Don't. Election Law Journal: Rules, Politics, and Policy (2020)
25. Lindeman, M., Halvorson, M., Smith, P., Garland, L., Addona, V., McCrea, D.: Principles and best practices for post-election audits (2008)
26. Lindeman, M., Stark, P., Yates, V.: BRAVO: Ballot-polling risk-limiting audits to verify outcomes. In: Proceedings of the 2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE '11). USENIX (2012)
27. Lindeman, M.: Rhode Island presidential risk-limiting audit, November 19-24, 2020 (brief report). https://elections.ri.gov/publications/Election_Publications/RLA/Rhode%20Island%20presidential%20RLA%20brief%20report.pdf
28. Lindeman, M., Halvorson, M., Smith, P., Garland, L., Addona, V., McCrea, D.: Principles and Best Practices for Post-Election Audits (Sep 2008), http://electionaudits.org/files/bestpracticesfinal_0.pdf
29. Lindeman, M., Stark, P.B.: A gentle introduction to risk-limiting audits. IEEE Security & Privacy **10**(5), 42–49 (2012)
30. McCarthy, J., Stanislevic, H., Lindeman, M., Ash, A., Addona, V., Batcher, M.: Percentage-based versus SAFE vote tabulation auditing: a graphic comparison. The American Statistician **62**(1), 11–16 (2008)
31. Michigan Secretary of State: Statewide risk-limiting election audit process to begin at 11 a.m. https://www.michigan.gov/som/0,4669,7-192-47796-549191--,00.html
32. Morin, S., McClearn, G., McBurnett, N., Vora, P.L., Zagórski, F.: A Note on Risk-Limiting Bayesian Polling Audits for Two-Candidate Elections (2020)
33. Ottoboni, K., Bernhard, M., Halderman, J.A., Rivest, R.L., Stark, P.B.: Bernoulli ballot polling: a manifest improvement for risk-limiting audits. In: Financial Cryptography and Data Security: FC 2019 International Workshops, VOTING and WTSC, St. Kitts, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers. pp. 226–241. Springer (2019)
34. Ottoboni, K., Stark, P.B., Lindeman, M., McBurnett, N.: Risk-Limiting Audits by Stratified Union-Intersection Tests of Elections (SUITE). In: International Joint Conference on Electronic Voting. pp. 174–188. Springer (2018)
35. Pennsylvania Secretary of State: Risk-Limiting Audit Pilot of November 2020 Presidential Election Finds Strong Evidence of Accurate Count. https://www.media.pa.gov/pages/State-details.aspx?newsid=453
36. Rhode Island Risk Limiting Audit Working Group: Pilot Implementation Study of Risk-Limiting Audit Methods in the State of Rhode Island, https://www.verifiedvoting.org/report-on-rhode-island-risk-limiting-audit-pilot-implementation-study-released/
37. Rivest, R.L., Shen, E.: A Bayesian method for auditing elections. In: USENIX Electronic Voting Technology Workshop / Workshop on Trustworthy Elections. EVT/WOTE '12 (Aug 2012), https://www.usenix.org/system/files/conference/evtwote12/rivest_bayes_rev_073112.pdf
38. Rivest, R.: On the notion of 'software independence' in voting systems. Phil. Trans. R. Soc. A **366**(1881), 3759–3767 (October 2008)

39. Rivest, R.: ClipAudit: A Simple Risk-Limiting Post-Election Audit (2017), https://arxiv.org/abs/1701.08312
40. Schürmann, C.: A risk-limiting audit in Denmark: A pilot. In: International Joint Conference on Electronic Voting. pp. 192–202. Springer (2016)
41. Sled, S.M.: Vertical proximity effects in the California Recall Election (2003)
42. Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., Halderman, J.A.: Security analysis of the Estonian internet voting system. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 703–715 (2014)
43. Sridhar, M., Rivest, R.L.: k-Cut: A Simple Approximately-Uniform Method for Sampling Ballots in Post-election Audits. In: International Conference on Financial Cryptography and Data Security. pp. 242–256. Springer (2019)
44. Stark, P.B.: Tools for Ballot-Polling Risk-Limiting Election Audits. https://www.stat.berkeley.edu/~stark/Vote/ballotPollTools.htm
45. Stark, P.B.: Tools for Comparison Risk-Limiting Election Audits . https://www.stat.berkeley.edu/~stark/Vote/auditTools.htm
46. Stark, P.B.: Conservative statistical post-election audits. Ann. Appl. Stat. **2**(2), 550–581 (2008)
47. Stark, P.B.: Election audits by sampling with probability proportional to an error bound: dealing with discrepancies (2008)
48. Stark, P.B.: CAST: Canvass audits by sampling and testing. IEEE Transactions on Information Forensics and Security **4**(4), 708–717 (2009)
49. Stark, P.B.: Efficient post-election audits of multiple contests: 2009 California tests. In: CELS 2009 4th annual conference on empirical legal studies paper (2009)
50. Stark, P.B.: Super-simple simultaneous single-ballot risk-limiting audits. In: Proceedings of the 2010 international conference on Electronic voting technology/workshop on trustworthy elections. pp. 1–16 (2010)
51. Stark, P.B.: Sets of half-average nulls generate risk-limiting audits: SHANGRLA. In: International Conference on Financial Cryptography and Data Security. pp. 319–336. Springer (2020)
52. Stark, P.B., Wagner, D.: Evidence-based elections. IEEE Security & Privacy **10**(5), 33–41 (2012)
53. Stein, R.M., Mann, C., Stewart III, C., Birenbaum, Z., Fung, A., Greenberg, J., Kawsar, F., Alberda, G., Alvarez, R.M., Atkeson, L., et al.: Waiting to vote in the 2016 presidential election: Evidence from a multi-county study. Political Research Quarterly **73**(2), 439–453 (2020)
54. The Cybersecurity and Infrastructure Security Agency (CISA): Leveraging the .gov Top-level Domain. https://www.cisa.gov/sites/default/files/publications/cisa-leveraging-the-gov-top-level-domain.pdf
55. Vora, P.L.: Risk-Limiting Bayesian Polling Audits for Two Candidate Elections. arXiv preprint arXiv:1902.00999 (2019)
56. VotingWorks: Arlo: Open-source risk-limiting audit software by VotingWorks. https://github.com/votingworks/arlo
57. Wolchok, S., Wustrow, E., Halderman, J.A., Prasad, H.K., Kankipati, A., Sakhamuri, S.K., Yagati, V., Gonggrijp, R.: Security analysis of India's electronic voting machines. In: Proceedings of the 17th ACM conference on Computer and communications security. pp. 1–14 (2010)
58. Zagórski, F., McClearn, G., Morin, S., McBurnett, N., Vora, P.L.: The Athena Class of Risk-Limiting Ballot Polling Audits. arXiv preprint arXiv:2008.02315 (2020)