SOBA: Secrecy-preserving Observable Ballot-level Audit

Josh Benaloh, Microsoft Research
Douglas Jones, Dept. of Computer Science, Univ. of Iowa
Eric L. Lazarus, DecisionSmith
Mark Lindeman
Philip B. Stark, Dept. of Statistics, Univ. of California, Berkeley

USENIX EVT/WOTE
San Francisco, CA
9 August 2011

What’s new here?

Way to audit that:

- Has a big chance of correcting the outcome if the outcome is wrong (risk-limiting).
- Enables the public to have strong evidence that the outcome is right, without having to trust (many) others.
- Preserves voter privacy.
- Is efficient, affordable, and currently feasible.

Motivation

- Risk-limiting audits now widely considered best practice.
- Auditing individual ballots requires least counting.
- Auditing individual ballots increases transparency.
- Simultaneously auditing all contests on each selected ballot can increase efficiency.
- Publishing data at the ballot level can compromise voter privacy.
- But if the raw data aren't published, public might not trust the results or the audit.
- Can we keep the benefits of simultaneous auditing at the ballot level and have data transparency without compromising privacy?
- E2E could do it, but requires changes, heavy crypto, "critical mass" of voters.
- Is there a bolt-on solution that doesn't require much change to voting systems or procedures, and that relies less on mathy stuff?

Motivation

- Risk-limiting audits now widely considered best practice.
- Auditing individual ballots requires least counting.
- Auditing individual ballots increases transparency.
- Simultaneously auditing all contests on each selected ballot can increase efficiency.
- Publishing data at the ballot level can compromise voter privacy.
- But if the raw data aren't published, public might not trust the results or the audit.
- Can we keep the benefits of simultaneous auditing at the ballot level and have data transparency without compromising privacy?
- E2E could do it, but requires changes, heavy crypto, "critical mass" of voters.
- Is there a bolt-on solution that doesn't require much change to voting systems or procedures, and that relies less on mathy stuff?

Motivation

- Risk-limiting audits now widely considered best practice.
- Auditing individual ballots requires least counting.
- Auditing individual ballots increases transparency.
- Simultaneously auditing all contests on each selected ballot can increase efficiency.
- Publishing data at the ballot level can compromise voter privacy.
- But if the raw data aren’t published, public might not trust the results or the audit.
- Can we keep the benefits of simultaneous auditing at the ballot level and have data transparency without compromising privacy?
- E2E could do it, but requires changes, heavy crypto, “critical mass” of voters.
- Is there a bolt-on solution that doesn’t require much change to voting systems or procedures, and that relies less on mathy stuff?

Motivation

- Risk-limiting audits now widely considered best practice.
- Auditing individual ballots requires least counting.
- Auditing individual ballots increases transparency.
- Simultaneously auditing all contests on each selected ballot can increase efficiency.
- Publishing data at the ballot level can compromise voter privacy.
- But if the raw data aren't published, public might not trust the results or the audit.
- Can we keep the benefits of simultaneous auditing at the ballot level and have data transparency without compromising privacy?
- E2E could do it, but requires changes, heavy crypto, "critical mass" of voters.
- Is there a bolt-on solution that doesn't require much change to voting systems or procedures, and that relies less on mathy stuff?

Motivation

- Risk-limiting audits now widely considered best practice.
- Auditing individual ballots requires least counting.
- Auditing individual ballots increases transparency.
- Simultaneously auditing all contests on each selected ballot can increase efficiency.
- Publishing data at the ballot level can compromise voter privacy.
- But if the raw data aren't published, public might not trust the results or the audit.
- Can we keep the benefits of simultaneous auditing at the ballot level and have data transparency without compromising privacy?
- E2E could do it, but requires changes, heavy crypto, "critical mass" of voters.
- Is there a bolt-on solution that doesn't require much change to voting systems or procedures, and that relies less on mathy stuff?

Motivation

- Risk-limiting audits now widely considered best practice.
- Auditing individual ballots requires least counting.
- Auditing individual ballots increases transparency.
- Simultaneously auditing all contests on each selected ballot can increase efficiency.
- Publishing data at the ballot level can compromise voter privacy.
- But if the raw data aren’t published, public might not trust the results or the audit.
- Can we keep the benefits of simultaneous auditing at the ballot level and have data transparency without compromising privacy?
- E2E could do it, but requires changes, heavy crypto, “critical mass” of voters.
- Is there a bolt-on solution that doesn’t require much change to voting systems or procedures, and that relies less on mathy stuff?

Motivation

- Risk-limiting audits now widely considered best practice.
- Auditing individual ballots requires least counting.
- Auditing individual ballots increases transparency.
- Simultaneously auditing all contests on each selected ballot can increase efficiency.
- Publishing data at the ballot level can compromise voter privacy.
- But if the raw data aren't published, public might not trust the results or the audit.
- Can we keep the benefits of simultaneous auditing at the ballot level and have data transparency without compromising privacy?
- E2E could do it, but requires changes, heavy crypto, "critical mass" of voters.
- Is there a bolt-on solution that doesn't require much change to voting systems or procedures, and that relies less on mathy stuff?

Motivation

- Risk-limiting audits now widely considered best practice.
- Auditing individual ballots requires least counting.
- Auditing individual ballots increases transparency.
- Simultaneously auditing all contests on each selected ballot can increase efficiency.
- Publishing data at the ballot level can compromise voter privacy.
- But if the raw data aren't published, public might not trust the results or the audit.
- Can we keep the benefits of simultaneous auditing at the ballot level and have data transparency without compromising privacy?
- E2E could do it, but requires changes, heavy crypto, "critical mass" of voters.
- Is there a bolt-on solution that doesn't require much change to voting systems or procedures, and that relies less on mathy stuff?

Motivation

- Risk-limiting audits now widely considered best practice.
- Auditing individual ballots requires least counting.
- Auditing individual ballots increases transparency.
- Simultaneously auditing all contests on each selected ballot can increase efficiency.
- Publishing data at the ballot level can compromise voter privacy.
- But if the raw data aren't published, public might not trust the results or the audit.
- Can we keep the benefits of simultaneous auditing at the ballot level and have data transparency without compromising privacy?
- E2E could do it, but requires changes, heavy crypto, "critical mass" of voters.
- Is there a bolt-on solution that doesn't require much change to voting systems or procedures, and that relies less on mathy stuff?

Definitions

- *Audit trail* or *ballot*: indelible record of how voters cast their votes, e.g., voter-marked paper ballot or VVPAT.
- *Outcome* of a contest: set of winners, not the exact vote counts.
- *Apparent outcome*: winner or winners according to the voting system.
- *Correct outcome*: winner or winners that a full hand count of the audit trail would find.
- Apparent outcome is *wrong* if it isn't the outcome a full hand count of the audit trail would show.

Definitions

- *Audit trail* or *ballot*: indelible record of how voters cast their votes, e.g., voter-marked paper ballot or VVPAT.
- *Outcome* of a contest: set of winners, not the exact vote counts.
- *Apparent outcome*: winner or winners according to the voting system.
- *Correct outcome*: winner or winners that a full hand count of the audit trail would find.
- Apparent outcome is *wrong* if it isn't the outcome a full hand count of the audit trail would show.

Definitions

- *Audit trail* or *ballot*: indelible record of how voters cast their votes, e.g., voter-marked paper ballot or VVPAT.
- *Outcome* of a contest: set of winners, not the exact vote counts.
- *Apparent outcome*: winner or winners according to the voting system.
- *Correct outcome*: winner or winners that a full hand count of the audit trail would find.
- Apparent outcome is *wrong* if it isn't the outcome a full hand count of the audit trail would show.

Definitions

- *Audit trail* or *ballot*: indelible record of how voters cast their votes, e.g., voter-marked paper ballot or VVPAT.
- *Outcome* of a contest: set of winners, not the exact vote counts.
- *Apparent outcome*: winner or winners according to the voting system.
- *Correct outcome*: winner or winners that a full hand count of the audit trail would find.
- Apparent outcome is *wrong* if it isn't the outcome a full hand count of the audit trail would show.

Definitions

- *Audit trail* or *ballot*: indelible record of how voters cast their votes, e.g., voter-marked paper ballot or VVPAT.
- *Outcome* of a contest: set of winners, not the exact vote counts.
- *Apparent outcome*: winner or winners according to the voting system.
- *Correct outcome*: winner or winners that a full hand count of the audit trail would find.
- Apparent outcome is *wrong* if it isn't the outcome a full hand count of the audit trail would show.

Risk-limiting audits

- *Risk-limiting audit*: pre-specified minimum chance of correcting apparent outcome if apparent outcome is wrong.
- *Risk*: largest possible chance an apparent outcome that's wrong won't be caught and corrected—no matter why it's wrong.
- *Simultaneous risk-limiting audit*: pre-specified minimum chance of correcting all incorrect apparent outcomes in the election.
- *Simultaneous risk*: largest possible chance that one or more wrong outcomes won't be caught and corrected—no matter why they are wrong.

Risk-limiting audits

- *Risk-limiting audit*: pre-specified minimum chance of correcting apparent outcome if apparent outcome is wrong.
- *Risk*: largest possible chance an apparent outcome that's wrong won't be caught and corrected—no matter why it's wrong.
- *Simultaneous risk-limiting audit*: pre-specified minimum chance of correcting all incorrect apparent outcomes in the election.
- *Simultaneous risk*: largest possible chance that one or more wrong outcomes won't be caught and corrected—no matter why they are wrong.

Risk-limiting audits

- *Risk-limiting audit*: pre-specified minimum chance of correcting apparent outcome if apparent outcome is wrong.
- *Risk*: largest possible chance an apparent outcome that's wrong won't be caught and corrected—no matter why it's wrong.
- *Simultaneous risk-limiting audit*: pre-specified minimum chance of correcting all incorrect apparent outcomes in the election.
- *Simultaneous risk*: largest possible chance that one or more wrong outcomes won't be caught and corrected—no matter why they are wrong.

Risk-limiting audits

- *Risk-limiting audit*: pre-specified minimum chance of correcting apparent outcome if apparent outcome is wrong.
- *Risk*: largest possible chance an apparent outcome that's wrong won't be caught and corrected—no matter why it's wrong.
- *Simultaneous risk-limiting audit*: pre-specified minimum chance of correcting all incorrect apparent outcomes in the election.
- *Simultaneous risk*: largest possible chance that one or more wrong outcomes won't be caught and corrected—no matter why they are wrong.

Compliance audit: check creation and curation of audit trail

- Did election use equipment that should create an accurate audit trail and adhere to procedures that should keep the audit trail sufficiently accurate to reflect the outcome according to how voters actually voted?
- Compliance audit should include ballot accounting, checks of seals, chain of custody, surveillance tapes, etc.
- If compliance audit generates convincing affirmative evidence that a full hand count of the audit trail would show the outcome according to how votes were cast, proceed to risk-limiting audit.
- If not, need a re-vote.

Compliance audit: check creation and curation of audit trail

- Did election use equipment that should create an accurate audit trail and adhere to procedures that should keep the audit trail sufficiently accurate to reflect the outcome according to how voters actually voted?
- Compliance audit should include ballot accounting, checks of seals, chain of custody, surveillance tapes, etc.
- If compliance audit generates convincing affirmative evidence that a full hand count of the audit trail would show the outcome according to how votes were cast, proceed to risk-limiting audit.
- If not, need a re-vote.

Compliance audit: check creation and curation of audit trail

- Did election use equipment that should create an accurate audit trail and adhere to procedures that should keep the audit trail sufficiently accurate to reflect the outcome according to how voters actually voted?
- Compliance audit should include ballot accounting, checks of seals, chain of custody, surveillance tapes, etc.
- If compliance audit generates convincing affirmative evidence that a full hand count of the audit trail would show the outcome according to how votes were cast, proceed to risk-limiting audit.
- If not, need a re-vote.

Compliance audit: check creation and curation of audit trail

- Did election use equipment that should create an accurate audit trail and adhere to procedures that should keep the audit trail sufficiently accurate to reflect the outcome according to how voters actually voted?
- Compliance audit should include ballot accounting, checks of seals, chain of custody, surveillance tapes, etc.
- If compliance audit generates convincing affirmative evidence that a full hand count of the audit trail would show the outcome according to how votes were cast, proceed to risk-limiting audit.
- If not, need a re-vote.

Goal of SOBA

Personally verifiable privacy-preserving *P*-resilient canvass framework.

WTF?

More Definitions

- *Canvass framework*: the vote-tabulation system together with other human, hardware, software, and procedural components of the canvass, including compliance audit and other audits.
- Canvass framework is *resilient with probability P* or *P-resilient* if the probability that the outcome it gives is the correct outcome is at least P, even if its software has an error, shortcoming, or undetected change: System tends to recover from (some) faults. (Strong software independence [Rivest & Wack], plus procedures that exploit that independence.)
- P-resilience can mean requiring a re-vote if the audit trail can't be shown to be in good shape.

More Definitions

- *Canvass framework*: the vote-tabulation system together with other human, hardware, software, and procedural components of the canvass, including compliance audit and other audits.
- Canvass framework is *resilient with probability P* or *P-resilient* if the probability that the outcome it gives is the correct outcome is at least P, even if its software has an error, shortcoming, or undetected change: System tends to recover from (some) faults. (Strong software independence [Rivest & Wack], plus procedures that exploit that independence.)
- P-resilience can mean requiring a re-vote if the audit trail can't be shown to be in good shape.

More Definitions

- *Canvass framework*: the vote-tabulation system together with other human, hardware, software, and procedural components of the canvass, including compliance audit and other audits.
- Canvass framework is *resilient with probability P* or *P-resilient* if the probability that the outcome it gives is the correct outcome is at least P, even if its software has an error, shortcoming, or undetected change: System tends to recover from (some) faults. (Strong software independence [Rivest & Wack], plus procedures that exploit that independence.)
- P-resilience can mean requiring a re-vote if the audit trail can't be shown to be in good shape.

and more . . .

- Canvass framework is *personally verifiable P-resilient* if it is *P*-resilient and a single individual could, as a practical matter, observe enough of the process to have convincing evidence that the canvass framework is in fact *P*-resilient.
- *Personally verifiable privacy-preserving P-resilient* canvass framework: personally verifiable *P*-resilient and it does not sacrifice privacy unnecessarily.

and more . . .

- Canvass framework is *personally verifiable P-resilient* if it is P-resilient and a single individual could, as a practical matter, observe enough of the process to have convincing evidence that the canvass framework is in fact P-resilient.
- *Personally verifiable privacy-preserving P-resilient* canvass framework: personally verifiable P-resilient and it does not sacrifice privacy unnecessarily.

Neither *personally verifiable* nor *privacy-preserving* is mathematically precise; *P*-resilience is.

“Personally verifiable” and “privacy-preserving” can be defined separately from “P-resilience.”

SOBA

- Adds a special risk-limiting audit to a strongly software-independent voting system that has had a compliance audit.
- Publishes results by ballot by contest: anybody can verify outcomes.
- Does not allow public to reconstruct whole-ballot CVRs, to protect privacy.
- Uses cryptographic commitment to allow auditors and observers to reconstruct the ballots selected for audit.
- Audit checks accuracy of CVRs *and* of the cryptographic commitment.

SOBA

- Adds a special risk-limiting audit to a strongly software-independent voting system that has had a compliance audit.
- Publishes results by ballot by contest: anybody can verify outcomes.
- Does not allow public to reconstruct whole-ballot CVRs, to protect privacy.
- Uses cryptographic commitment to allow auditors and observers to reconstruct the ballots selected for audit.
- Audit checks accuracy of CVRs *and* of the cryptographic commitment.

SOBA

- Adds a special risk-limiting audit to a strongly software-independent voting system that has had a compliance audit.
- Publishes results by ballot by contest: anybody can verify outcomes.
- Does not allow public to reconstruct whole-ballot CVRs, to protect privacy.
- Uses cryptographic commitment to allow auditors and observers to reconstruct the ballots selected for audit.
- Audit checks accuracy of CVRs *and* of the cryptographic commitment.

SOBA

- Adds a special risk-limiting audit to a strongly software-independent voting system that has had a compliance audit.
- Publishes results by ballot by contest: anybody can verify outcomes.
- Does not allow public to reconstruct whole-ballot CVRs, to protect privacy.
- Uses cryptographic commitment to allow auditors and observers to reconstruct the ballots selected for audit.
- Audit checks accuracy of CVRs *and* of the cryptographic commitment.

SOBA

- Adds a special risk-limiting audit to a strongly software-independent voting system that has had a compliance audit.
- Publishes results by ballot by contest: anybody can verify outcomes.
- Does not allow public to reconstruct whole-ballot CVRs, to protect privacy.
- Uses cryptographic commitment to allow auditors and observers to reconstruct the ballots selected for audit.
- Audit checks accuracy of CVRs *and* of the cryptographic commitment.

Aside: cryptographic commitments

- Ensures that the ballot identifier is secret but indelible, so every ballot is properly reflected in the electronic results.
- Select and publish commitment function $H()$.
- To commit that a given CCVR comes from ballot b, LEO selects secret “salt” u and computes $y = H(b, u)$. Publishes shrouded ID (SID) y.
- If ballot b is selected for audit, LEO can reveal u and b: Anyone can check whether $y = H(b, u)$.

Aside: cryptographic commitments

- Ensures that the ballot identifier is secret but indelible, so every ballot is properly reflected in the electronic results.
- Select and publish commitment function $H()$.
- To commit that a given CCVR comes from ballot b, LEO selects secret “salt” u and computes $y = H(b, u)$. Publishes shrouded ID (SID) y.
- If ballot b is selected for audit, LEO can reveal u and b: Anyone can check whether $y = H(b, u)$.

Aside: cryptographic commitments

- Ensures that the ballot identifier is secret but indelible, so every ballot is properly reflected in the electronic results.
- Select and publish commitment function $H()$.
- To commit that a given CCVR comes from ballot b, LEO selects secret "salt" u and computes $y = H(b, u)$. Publishes shrouded ID (SID) y.
- If ballot b is selected for audit, LEO can reveal u and b: Anyone can check whether $y = H(b, u)$.

Aside: cryptographic commitments

- Ensures that the ballot identifier is secret but indelible, so every ballot is properly reflected in the electronic results.
- Select and publish commitment function $H()$.
- To commit that a given CCVR comes from ballot b, LEO selects secret "salt" u and computes $y = H(b, u)$. Publishes shrouded ID (SID) y.
- If ballot b is selected for audit, LEO can reveal u and b: Anyone can check whether $y = H(b, u)$.

Commitment function key properties: *binding* (*collision-resistant*), and *hiding* (*one-way*).

- *Binding*: infeasible to find any pair $(b', u') \neq (b, u)$ for which $H(b', u') = H(b, u)$. Helps ensure nobody can claim more than one CCVR for a given contest comes from the same ballot.
- *Hiding*: infeasible for anyone with access only to the SIDs to learn anything about which ballot is involved in each commitment.

Salt should be random number with at least 128 digits.

Commitment function key properties: *binding* (*collision-resistant*), and *hiding* (*one-way*).

- *Binding*: infeasible to find any pair $(b', u') \neq (b, u)$ for which $H(b', u') = H(b, u)$. Helps ensure nobody can claim more than one CCVR for a given contest comes from the same ballot.
- *Hiding*: infeasible for anyone with access only to the SIDs to learn anything about which ballot is involved in each commitment.

Salt should be random number with at least 128 digits.

SOBA preparations

C contests, N_c ballots cast in contest c, N ballots in all, M voting opportunities in all.

- Compliance audit, including ballot accounting: determine $\{N_c\}$, N, M.
- Find apparent outcomes of the C contests.
- Construct CVR for each ballot (perhaps by unofficial scan: transitive auditing); assign unique ID to each ballot.
- Disaggregate CVRs into C per-contest sets of *CCVRs*; Publish C CCVR files. N_c lines in file c, each gives CCVR and SID. Sort by SID.
- Publish *ballot style file*. N lines. Each line lists contests on ballot and a unique ballot ID (e.g., #17,097, or 275th in 39th deck).
- Construct (but don't publish) *lookup file*. M lines, 3 entries per line: SID, corresponding unshrouded ID b, and "salt" u
- Select and disclose H, risk limit, PRNG.

SOBA preparations

C contests, N_c ballots cast in contest c, N ballots in all, M voting opportunities in all.

- Compliance audit, including ballot accounting: determine $\{N_c\}$, N, M.
- Find apparent outcomes of the C contests.
- Construct CVR for each ballot (perhaps by unofficial scan: transitive auditing); assign unique ID to each ballot.
- Disaggregate CVRs into C per-contest sets of *CCVRs*; Publish C CCVR files. N_c lines in file c, each gives CCVR and SID. Sort by SID.
- Publish *ballot style file*. N lines. Each line lists contests on ballot and a unique ballot ID (e.g., #17,097, or 275th in 39th deck).
- Construct (but don't publish) *lookup file*. M lines, 3 entries per line: SID, corresponding unshrouded ID b, and "salt" u
- Select and disclose H, risk limit, PRNG.

SOBA preparations

C contests, N_c ballots cast in contest c, N ballots in all, M voting opportunities in all.

- Compliance audit, including ballot accounting: determine $\{N_c\}$, N, M.
- Find apparent outcomes of the C contests.
- Construct CVR for each ballot (perhaps by unofficial scan: transitive auditing); assign unique ID to each ballot.
- Disaggregate CVRs into C per-contest sets of *CCVRs*; Publish C CCVR files. N_c lines in file c, each gives CCVR and SID. Sort by SID.
- Publish *ballot style file*. N lines. Each line lists contests on ballot and a unique ballot ID (e.g., #17,097, or 275th in 39th deck).
- Construct (but don't publish) *lookup file*. M lines, 3 entries per line: SID, corresponding unshrouded ID b, and "salt" u
- Select and disclose H, risk limit, PRNG.

SOBA preparations

C contests, N_c ballots cast in contest c, N ballots in all, M voting opportunities in all.

- Compliance audit, including ballot accounting: determine $\{N_c\}$, N, M.
- Find apparent outcomes of the C contests.
- Construct CVR for each ballot (perhaps by unofficial scan: transitive auditing); assign unique ID to each ballot.
- Disaggregate CVRs into C per-contest sets of *CCVRs*; Publish C CCVR files. N_c lines in file c, each gives CCVR and SID. Sort by SID.
- Publish *ballot style file*. N lines. Each line lists contests on ballot and a unique ballot ID (e.g., #17,097, or 275th in 39th deck).
- Construct (but don't publish) *lookup file*. M lines, 3 entries per line: SID, corresponding unshrouded ID b, and "salt" u
- Select and disclose H, risk limit, PRNG.

SOBA preparations

C contests, N_c ballots cast in contest c, N ballots in all, M voting opportunities in all.

- Compliance audit, including ballot accounting: determine $\{N_c\}$, N, M.
- Find apparent outcomes of the C contests.
- Construct CVR for each ballot (perhaps by unofficial scan: transitive auditing); assign unique ID to each ballot.
- Disaggregate CVRs into C per-contest sets of *CCVRs*; Publish C CCVR files. N_c lines in file c, each gives CCVR and SID. Sort by SID.
- Publish *ballot style file*. N lines. Each line lists contests on ballot and a unique ballot ID (e.g., #17,097, or 275th in 39th deck).
- Construct (but don't publish) *lookup file*. M lines, 3 entries per line: SID, corresponding unshrouded ID b, and "salt" u
- Select and disclose H, risk limit, PRNG.

SOBA preparations

C contests, N_c ballots cast in contest c, N ballots in all, M voting opportunities in all.

- Compliance audit, including ballot accounting: determine $\{N_c\}$, N, M.
- Find apparent outcomes of the C contests.
- Construct CVR for each ballot (perhaps by unofficial scan: transitive auditing); assign unique ID to each ballot.
- Disaggregate CVRs into C per-contest sets of *CCVRs*; Publish C CCVR files. N_c lines in file c, each gives CCVR and SID. Sort by SID.
- Publish *ballot style file*. N lines. Each line lists contests on ballot and a unique ballot ID (e.g., #17,097, or 275th in 39th deck).
- Construct (but don't publish) *lookup file*. M lines, 3 entries per line: SID, corresponding unshrouded ID b, and "salt" u
- Select and disclose H, risk limit, PRNG.

SOBA preparations

C contests, N_c ballots cast in contest c, N ballots in all, M voting opportunities in all.

- Compliance audit, including ballot accounting: determine $\{N_c\}$, N, M.
- Find apparent outcomes of the C contests.
- Construct CVR for each ballot (perhaps by unofficial scan: transitive auditing); assign unique ID to each ballot.
- Disaggregate CVRs into C per-contest sets of *CCVRs*; Publish C CCVR files. N_c lines in file c, each gives CCVR and SID. Sort by SID.
- Publish *ballot style file*. N lines. Each line lists contests on ballot and a unique ballot ID (e.g., #17,097, or 275th in 39th deck).
- Construct (but don't publish) *lookup file*. M lines, 3 entries per line: SID, corresponding unshrouded ID b, and "salt" u
- Select and disclose H, risk limit, PRNG.

What can go wrong?

The CCVRs might fail to be sufficiently accurate because

- At least one CCVR and the ballot it purports to represent do not match because human and machine interpretations of voter intent differ (for instance, because the voter marked the ballot improperly). This is a failure of the generation of CCVRs.
- At least one CCVR does not in fact correspond to any ballot. It is an "orphan." This is a failure of the mapping between ballots and CCVRs.
- More than one CCVR for the same contest is mapped to the same ballot. It is a "multiple." This is also a failure of the mapping between ballots and CCVRs.
- There is no CCVR corresponding to some voting opportunity on a ballot.

Audit checks these things *while* checking the accuracy of the CCVRs, with the same sample.

What can go wrong?

The CCVRs might fail to be sufficiently accurate because

- At least one CCVR and the ballot it purports to represent do not match because human and machine interpretations of voter intent differ (for instance, because the voter marked the ballot improperly). This is a failure of the generation of CCVRs.
- At least one CCVR does not in fact correspond to any ballot. It is an "orphan." This is a failure of the mapping between ballots and CCVRs.
- More than one CCVR for the same contest is mapped to the same ballot. It is a "multiple." This is also a failure of the mapping between ballots and CCVRs.
- There is no CCVR corresponding to some voting opportunity on a ballot.

Audit checks these things *while* checking the accuracy of the CCVRs, with the same sample.

What can go wrong?

The CCVRs might fail to be sufficiently accurate because

- At least one CCVR and the ballot it purports to represent do not match because human and machine interpretations of voter intent differ (for instance, because the voter marked the ballot improperly). This is a failure of the generation of CCVRs.
- At least one CCVR does not in fact correspond to any ballot. It is an “orphan.” This is a failure of the mapping between ballots and CCVRs.
- More than one CCVR for the same contest is mapped to the same ballot. It is a “multiple.” This is also a failure of the mapping between ballots and CCVRs.
- There is no CCVR corresponding to some voting opportunity on a ballot.

Audit checks these things *while* checking the accuracy of the CCVRs, with the same sample.

What can go wrong?

The CCVRs might fail to be sufficiently accurate because

- At least one CCVR and the ballot it purports to represent do not match because human and machine interpretations of voter intent differ (for instance, because the voter marked the ballot improperly). This is a failure of the generation of CCVRs.
- At least one CCVR does not in fact correspond to any ballot. It is an "orphan." This is a failure of the mapping between ballots and CCVRs.
- More than one CCVR for the same contest is mapped to the same ballot. It is a "multiple." This is also a failure of the mapping between ballots and CCVRs.
- There is no CCVR corresponding to some voting opportunity on a ballot.

Audit checks these things *while* checking the accuracy of the CCVRs, with the same sample.

SOBA Audit at 10% risk limit

1. Verify that, for each contest c, there are N_c entries in the CCVR file for contest c.
2. Verify that, for each contest c, the CCVR file shows the same outcome (not count!) as the reported outcome. If not, hand count any discrepant contests.
3. Verify that the $M = N_1 + \cdots + N_C$ shrouded ballot identifiers in all C CCVR files are unique.

SOBA Audit at 10% risk limit

1. Verify that, for each contest c, there are N_c entries in the CCVR file for contest c.
2. Verify that, for each contest c, the CCVR file shows the same outcome (not count!) as the reported outcome. If not, hand count any discrepant contests.
3. Verify that the $M = N_1 + \cdots + N_C$ shrouded ballot identifiers in all C CCVR files are unique.

SOBA Audit at 10% risk limit

1. Verify that, for each contest c, there are N_c entries in the CCVR file for contest c.
2. Verify that, for each contest c, the CCVR file shows the same outcome (not count!) as the reported outcome. If not, hand count any discrepant contests.
3. Verify that the $M = N_1 + \cdots + N_C$ shrouded ballot identifiers in all C CCVR files are unique.

4. Verify that, for each contest c, there are N_c entries in the ballot style file that list the contest.
5. Verify that the ballot identifiers in the ballot style file are unique.

If 1, 3, 4, or 5 fails, LEO needs to correct before risk-limiting stage of audit can start.

4. Verify that, for each contest c, there are N_c entries in the ballot style file that list the contest.
5. Verify that the ballot identifiers in the ballot style file are unique.

If 1, 3, 4, or 5 fails, LEO needs to correct before risk-limiting stage of audit can start.

6. Set audit parameters:
   6.1. Find *diluted margin* from CCVRs: smallest apparent margin in votes for any contest, divided by N.
   6.2. Set initial sample size $n = 7/(\text{diluted margin})$.
   6.3. Select a *seed* s. Observers could contribute to s or roll dice.
7. Select n pseudo-random numbers between 1 and N. Find those rows in the ballot style file. Retrieve corresponding ballots. Compare CVR with ballot for all contests on the ballot. If ballot has a contest the style file doesn't show, treat CCVR as vote for apparent winner. If style file says ballot has a contest ballot doesn't, treat ballot as vote for runner-up.
8. If no CVR in the sample overstated any margin by 2 votes, and fraction of CVRs that overstate any margin by one vote is at most 20% of the diluted margin, stop auditing.
9. Else, calculate the Kaplan-Markov P-value, P_{KM} (just multiplication & subtraction). Keep auditing until $P_{KM} \leq 10\%$ (then the audit stops) or you give up (and count remaining votes by hand).

6. Set audit parameters:
   6.1. Find *diluted margin* from CCVRs: smallest apparent margin in votes for any contest, divided by N.
   6.2. Set initial sample size $n = 7/$(diluted margin).
   6.3. Select a *seed* s. Observers could contribute to s or roll dice.
7. Select n pseudo-random numbers between 1 and N. Find those rows in the ballot style file. Retrieve corresponding ballots. Compare CVR with ballot for all contests on the ballot. If ballot has a contest the style file doesn't show, treat CCVR as vote for apparent winner. If style file says ballot has a contest ballot doesn't, treat ballot as vote for runner-up.
8. If no CVR in the sample overstated any margin by 2 votes, and fraction of CVRs that overstate any margin by one vote is at most 20% of the diluted margin, stop auditing.
9. Else, calculate the Kaplan-Markov P-value, P_{KM} (just multiplication & subtraction). Keep auditing until $P_{KM} \leq 10\%$ (then the audit stops) or you give up (and count remaining votes by hand).

6. Set audit parameters:
    - 6.1. Find *diluted margin* from CCVRs: smallest apparent margin in votes for any contest, divided by N.
    - 6.2. Set initial sample size $n = 7/(\text{diluted margin})$.
    - 6.3. Select a *seed* s. Observers could contribute to s or roll dice.
7. Select n pseudo-random numbers between 1 and N. Find those rows in the ballot style file. Retrieve corresponding ballots. Compare CVR with ballot for all contests on the ballot. If ballot has a contest the style file doesn't show, treat CCVR as vote for apparent winner. If style file says ballot has a contest ballot doesn't, treat ballot as vote for runner-up.
8. If no CVR in the sample overstated any margin by 2 votes, and fraction of CVRs that overstate any margin by one vote is at most 20% of the diluted margin, stop auditing.
9. Else, calculate the Kaplan-Markov P-value, P_{KM} (just multiplication & subtraction). Keep auditing until $P_{KM} \leq 10\%$ (then the audit stops) or you give up (and count remaining votes by hand).

6. Set audit parameters:
   6.1. Find *diluted margin* from CCVRs: smallest apparent margin in votes for any contest, divided by N.
   6.2. Set initial sample size $n = 7/(\text{diluted margin})$.
   6.3. Select a *seed* s. Observers could contribute to s or roll dice.
7. Select n pseudo-random numbers between 1 and N. Find those rows in the ballot style file. Retrieve corresponding ballots. Compare CVR with ballot for all contests on the ballot. If ballot has a contest the style file doesn't show, treat CCVR as vote for apparent winner. If style file says ballot has a contest ballot doesn't, treat ballot as vote for runner-up.
8. If no CVR in the sample overstated any margin by 2 votes, and fraction of CVRs that overstate any margin by one vote is at most 20% of the diluted margin, stop auditing.
9. Else, calculate the Kaplan-Markov P-value, P_{KM} (just multiplication & subtraction). Keep auditing until $P_{KM} \leq 10\%$ (then the audit stops) or you give up (and count remaining votes by hand).

Missing pieces

- Methods for extracting CVRs quickly and reliably (more this pm).
- Ways to audit $\{N_c\}$ (more this pm).
- Good explanations of statistics and crypto: public relations.
- Best practices for creating & curating audit trail.
- Compliance audits.

Missing pieces

- Methods for extracting CVRs quickly and reliably (more this pm).
- Ways to audit $\{N_c\}$ (more this pm).
- Good explanations of statistics and crypto: public relations.
- Best practices for creating & curating audit trail.
- Compliance audits.

Missing pieces

- Methods for extracting CVRs quickly and reliably (more this pm).
- Ways to audit $\{N_c\}$ (more this pm).
- Good explanations of statistics and crypto: public relations.
- Best practices for creating & curating audit trail.
- Compliance audits.

Missing pieces

- Methods for extracting CVRs quickly and reliably (more this pm).
- Ways to audit $\{N_c\}$ (more this pm).
- Good explanations of statistics and crypto: public relations.
- Best practices for creating & curating audit trail.
- Compliance audits.

Missing pieces

- Methods for extracting CVRs quickly and reliably (more this pm).
- Ways to audit $\{N_c\}$ (more this pm).
- Good explanations of statistics and crypto: public relations.
- Best practices for creating & curating audit trail.
- Compliance audits.

Proof: 7 cases

1. The ballot style file has more than one entry that corresponds to the same actual ballot, or more than one actual ballot corresponds to the same entry in the ballot style file. Precluded by the uniqueness of the IDs and of the recipes for locating the actual ballot with each ID.
2. More than one ID corresponds to the same SID (for different values of u). Precluded by the binding property of H.
3. The ballot style file contains IDs that do not correspond to actual ballots, or claims that a ballot contains a contest that it does not actually contain. The biggest effect this could have on an apparent contest outcome is if the ballot that entry is supposed to match showed a vote for the runner-up in every missing contest, which is no greater than a two-vote change to any margin. Because the audit samples entries of the ballot style file with equal probability, this kind of error in an entry is just as likely to be revealed as any other. If such a ballot style file entry is selected for audit, step 7 treats it in a worst-case way.

4. The ballot style file claims that a ballot does not contain a contest that it does contain. The biggest effect this could have on an apparent contest outcome is if the CCVR for that contest showed a vote for the apparent winner, which cannot change the margin by more than two votes, so the error-bound assumptions are satisfied. Because the audit samples entries of the ballot style file with equal probability, this kind of error in an entry is just as likely to be revealed as any other. If such a ballot style file entry is selected for audit, step 7 treats it this worst-case way.
5. There are ballots whose IDs do not appear in the ballot style file. Since there are the same number of ballots as entries in the ballot style file and the IDs in the ballot style file are unique, there must be ballot identifiers in the ballot style file that do not match any ballot. Hence, case (3) holds.

6. There are CCVRs for which the SID is not the ID of any ballot. If the SID matches an ID in the ballot style file, we are in case (3). Suppose therefore that the SID does not match any in the ballot style file. Suppose this happens for contest *c*. The preliminary checks show that the ballot style file has exactly N_c entries for contest *c* and that there are exactly N_c entries in the CCVR file for contest *c*. Therefore, if there is such a CCVR, one of the ballot style file entries that lists contest *c* has an ID that does not occur as an SID in the CCVR file for that contest. The largest effect this could have on contest *c* is if the "substituted" CCVR entry reported a vote for the apparent winner; this cannot overstate the margin by more than two votes, so the audit's error-bound assumption still holds. Because the audit samples entries of the ballot style file with equal probability, this kind of error in a ballot style file entry is just as likely to be revealed as any other. If such a ballot style file entry is selected for audit, step 7 treats it this worst-case way.
7. The same ID appears in shrouded form more than once in a single CCVR file. As in the previous case, we know there are N_c entries in the CCVR file for contest *c* and N_c entries in the ballot style file that include contest *c*; moreover, the IDs in the ballot style file are unique. Hence, there must be at least one entry in the ballot style file that lists contest *c* for which the ID does not appear as an SID in the CCVR file. We are therefore in case (6).