SCITT: Election Data Use Case

Ray Lutz, Citizens Oversight -- raylutz@citizensoversight.org
V1 2019 -- Original Hazards analysis
Googledoc created 2022-05-06 for community comments and enhancement
Changed name and added background for SCITT submission -- 2022-10-16

A working group has formed under the umbrella of the Internet Engineering Task Force (IETF) to formulate standards to improve the security of supply chains, and in particular the security and trustworthiness of software. The working group is entitled Supply Chain Integrity, Transparency and Trust (SCITT). This infrastructure may be a good match for securing election data. This document summarizes the operational constraints of this use case.

See:

- https://github.com/ietf-scitt/charter/blob/master/ietf-scitt-charter.md -- SCITT Charter
- https://www.ietf.org/archive/id/draft-birkholz-scitt-architecture-00.txt -- Draft architecture

Election systems today, particularly in large districts, use paper ballots (either hand-marked or machine marked) and electronic scanners, which make images of those paper ballots and then process those images to evaluate voter intent. Most of the population lives in large districts where it is infeasible to count ballots by hand. Half of the U.S. voting population lives in only 175 of the most populous counties, while 60% lives in the most populous 300 and 70% lives in only 500 counties. The other 30% live in the other 4000 counties.

Data from election systems is still conformant only to proprietary formats and publication is on an ad hoc basis, generally not secured with hash values and it is difficult to prove that the data is not tampered with and that it represents the official results.

Elections stress systems (both computers and procedural) to the extreme due to the need to 1) fully identify voters, 2) then support private voting and ballot anonymity, 3) conduct the elections so that even insiders cannot manipulate the outcome, and 4) maintain an audit trail that documents verified voter intent and facilitates independent post-election audits.

Elections are more complex and difficult than the public generally understands. In a given county, the number of contests can be in the hundreds and the number of different ballot styles (combination of contests, languages, and rotating candidate order) in the thousands. As soon as computer technology became available, it was quickly adopted to help election officials deal with this complexity. The Help America Vote Act (HAVA) after the 2000 election debacle resulted in the deployment of "Direct Recording Electronic" (DRE) systems that proved to be unreliable and easily manipulated, while being impossible to fully audit. Now, most districts have adopted a paper record that can be voter-verified prior to casting, and this is the reason the 2020 election

was touted the most secure in history, while it is also true that we still have a lot to do. Security can never be perfect.

SCITT is envisioned as a number of "building blocks" that will allow arbitrary artifacts to be secured using existing public-key infrastructure (PKI) and to allow submission of any arbitrary data artifact, to validate the submitting entity and ensure that any change in the artifact can be detected. The functionality envisions for SCITT seems to be a good match to what is needed to further secure the election data and infrastructure.

Of course, for those election systems that are based on software and hardware systems, SCITT should also be deployable for improving software security, ballot scanners, and tabulators. Additionally, election results and supporting data can be published and then secured to prevent any alteration.

There are many places where elections exhibit hazards, and there are a variety of scams and frauds that can occur. Generally speaking, the use case is similar to the "Trusted Document Scanning" use case, but it is a more hostile environment where it is difficult to establish trusted insiders. Instead, transparent procedures and oversight by the public is relied upon as the only truly trusted party is the general public.

Most of the hazards of the election system are procedural in nature and are not necessarily amenable to treatment by SCITT and are beyond the scope of this analysis. Nevertheless, a hazard analysis is provided in Appendix 1.

The general Trusted Document Scanning use case is based on standards and guidelines that may mention digital signatures but they are not fully developed into a comprehensive system for cryptographic security. For example:

See

- "Federal Agencies Digital Guidelines Initiative (FADGI) -- Technical Guidelines for Digitizing Cultural Heritage Materials Creation of Raster Image Files https://www.digitizationguidelines.gov/guidelines/FADGI%20Federal%20%20Agencies%20Digital%20Guidelines%20Initiative-2016%20Final_rev1.pdf

- Trusted Digital Repositories: Attributes and Responsibilities https://www.oclc.org/content/dam/research/activities/trustedrep/repositories.pdf

- Recommendation for Space Data System Practices -- REFERENCE MODEL FOR AN OPEN ARCHIVAL INFORMATION SYSTEM (OAIS) https://public.ccsds.org/pubs/650x0m2.pdf

Focus for purposes of SCITT

America's Election Model: The Architecture of Elections --
https://pages.nist.gov/ElectionModeling/ElectionProcessModel.pdf

Voting System Software

Voting system software is certified by the Election Assistance Commission and there is a requirement for software / firmware to be submitted to the National Software Reference Library and signed using FIPS 140-2. But there is no way for the public to use this information to verify the software or firmware, nor is there any transparent and standardized control of the digital artifacts that result.

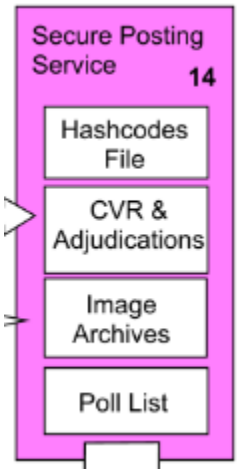
Voting systems are controlled by the Election Assistance Commission and the Voluntary Voting Systems Guidelines.
EAC VVSG 2.0 - https://www.eac.gov/voting-equipment/voluntary-voting-system-guidelines

Voting Software Reference Data Set -- this provides the currently certified software
https://www.nist.gov/itl/ssd/software-quality-group/voting-software-reference-data-set

Security of the software used in election systems is not unlike the needs for other software systems and therefore this document will instead focus on election data which has some special characteristics.

Data Artifact Security

The U.S. Election system has very limited controls of data artifacts. We can focus on the data artifacts and list some of the special requirements.



Regarding data produced, Block 14 in Hazards flow chart (in Appendix 1) is combined with the registered voters list and list of voters who voted from Block 4. This represents the data produced by the voting system and includes the artifacts to be secured.

These data are inputs into an optional post-election audit process, which also produces some artifacts so it can be also checked through public oversight. For Risk Limiting (statistical sampling audits) and other paper-based auditing procedures, they may also produce scans of tally sheets and of course the audit report.

At this stage, we can list some of the technical constraints:

1. Election offices are generally "air-gapped" and do not have access to the internet. Frequently, the results are placed on thumbdrives and "sneakernet" is used to transfer

the data either to consumers of the data (auditors) or to place it eventually on a file sharing service for public access.

2. Voter-facing scanners which may exist in polling places typically provide feedback to the voter to reduce error in completing the ballot. These machines typically today are also ballot aware. These machines are configured using a flash memory device (AKA thumbdrive) and these same devices provide aggregated results from the machine as well as ballot images and the CVR (cast vote record), which is a ballot-by-ballot interpretation of voter intent.

   NIST has a "Common Data Format" (CDF) standard for the Cast Vote Record (CVR) but it is largely not yet implemented by the vendors. Dominion has the closest implementation that is used in some equipment and only optionally, and it is not an exact implementation. Other vendors use non-standard formats specific to each vendor.

   (NIST) Special Publication 1500-103, Cast Vote Records Common Data Format Specification Version 1 -- https://pages.nist.gov/CastVoteRecords/

   For purposes of the SCITT use case, the exact data format can be ignored.

3. The CVR, any aggregated results, and tracking data are returned when the flash device is hand-carried back to the central office. Some states allow cellular modem to be used to return initial aggregated results followed by return of the flash devices to obtain full data. Vendors claim to encrypt the data on the flash devices, digitally signed using the private key of the voting machine. The mechanism for remote attestation of the machine is ad-hoc or nonexistent.

4. Eventually, the data can be considered as a whole. But there is some benefit to being able to produce cryptographic tracking (hash values) for batches that are not yet public. This will allow the files to be cryptographically secured to prohibit alteration while also not revealing the data until after the election. This suggests that the security certificate must be separated from the data (and thus the data must not be encapsulated in a payload).

5. Voter-facing voting machines may have an integrated scanner, or hand-marked paper ballots may be imaged by central scanning operations to produce a set of ballot images, typically grouped into precincts or mixed-precinct batches. Precincts and batches may average perhaps 200 ballot sheets each (imaged on both sides), but may range from 1 to 1000s. Eventually, these are grouped into ballot image archives (i.e. ZIP files).

6. The ballot image archive(s) may be very large with numerous small image files, average is about 300KB each, but some rescans are 1.5MB each. For ease of handling, these are placed into ZIP archives, with up to about 50K ballots in each one. A good handling size right now is less than 10GB per file, to make uploads feasible without too much time

spent on each one, in case there is a need for a retry. Thus, there is a need for a number of archives, perhaps a few dozen. In the case of non-voting system rescans, they are sometimes provided without a ZIP archive to make handling easier.

   a. The Clear Ballot rescans for Hillsborough County, FL (Tampa area) consumed about 2.6TB, for 717K ballots cast. These were individual JPEG files (unzipped), one per side (2 or more per ballot). These could be ZIP archived and would be easier to handle but would not further compress.

   b. The 2020 general election in Volusia, FL had 34 ZIP archives with a total of 123GB, making each one just under 4GB on the average, for 309K ballots cast.

   c. For Maricopa County AZ (second largest county nationally) there were 2.1 million ballot images, about 630GB. But can easily be placed on one 1TB thumbdrive.

7. Cast-Vote Records (CVRs) are the voting system evaluation of the vote cast on each ballot sheet. These may be multiple .xlsx files (which are internally zip compressed), with 99,999 records each (Election Systems & Software equipment, "ES&S") or may be many JSON files, typically one per scanner/tabulator, all in an ZIP archive (Dominion Voting Systems). The CVRs tend not to be too large.

   a. Volusia FL had 4 .xlsx files, totalling less than 100MB for 309K ballots cast.

   b. Bartow, GA had about 50K ballots and the single JSON chunk was 255MB but in a ZIP archive it consumed under 10MB.

   c. Maricopa County, AZ CVR consumed about 2GB in a ZIP archive.

8. There are also a number of other files that are commonly included, and all are not problematic in terms of being too numerous or too large. The only exception is when there are scans of tally sheets, for example, but are never as challenging as the ballot images themselves. (Maricopa County had 3 tally sheets for each of the 10,341 batches).

9. It would be preferred to produce a hash of the ballot image file as soon as it is produced in the scanner and digitally sign it with a self-generated private key for that scanner, to limit the exposure of the image to changes. Unfortunately, at this time there is no requirement for a security module which would have the capability of generating a private/public key pair using an internal noise source.

10. The above is to hopefully disallow the "unclear ballot" hack which has been described by the paper "Unclear Ballot" https://mbernhard.com/papers/unclearballot.pdf. See also this critical review of the paper: https://copswiki.org/Common/M1976

11. The set of all ballot images for typical "large" counties are large and therefore, the images themselves cannot (probably) be included in any SCITT payload. However, a data descriptor block may be all we need. Such a data descriptor can probably be used by just about any application. It must have the following attributes:
    a. A collection of collections of individual file descriptors.
        i. The outside collection is specific to the jurisdiction (typ. County).
        ii. The inside collections would represent a specific type of data.
        iii. Inside those, are the data artifacts, which relate to what is stored.
            1. The data artifacts may be ZIP files which may be further grouped into batches of images and hopefully a related security certificate.

    b. Able to be independent of any SCITT transparency service, but able to be checked using the SCITT service.
    c. As long as the ballot images are archived into ZIP archives to reduce the size of the list, the list is reasonable, say in the 10s or 100s. Otherwise, just listing the files is difficult, with about 1.5 million files in the case of Hillsborough county.

    d. The outside collection simply combines all file types in one jurisdiction for one election, i.e. like the files for one product.

    e. Internat collections include files of one type, such as ballot images, which are combined into chunks, and each chunk can be listed as a single item.

    f. Descriptor is the only thing that needs to be controlled by SCITT. The entire block need not be included in SCITT, only the hash of the block.

12. The functionality of SCITT appears to be mainly to combine three things:
    a. The official identity of the entity submitting,
    b. The logical name of the submission.
    c. A hash of the item.

13. For the purposes of this use case, perhaps it is best to allow a more complex list of items and their hashes grouped into groups of items in each type, and a list of those groups, one for each logical file type. Thus, we would have the ballot images, CVRs, poll lists, etc. and each item within those groups a list of ZIP files which correspond to the data items which are actually stored. It appears there will need to be a separate artifact which is a set of certificates that relate to batches which are individually secured that exist inside the ZIP archives. These certificates were likely generated as the ballot images were created by the individual voting machines or by central scanning operations as individual batches are scanned. (This functionality of voting machine scanners and central scanning operations is not available at this time but is envisioned as a requirement to thwart the unclear ballot hack).

14. It does not appear to be required that the item be included as a payload. There can be separate "repository" functionality apart from the SCITT tracking of these security blocks. To the extent the entity is trusted to begin with, it appears to be reasonable to also trust it to provide the hash of the object.

See also this analysis for securing voting scanners and ballot images.
https://copswiki.org/Common/M1936 Securing Digital Ballot Images to Enable Auditing

DBOM - Data BOM

For applications of this type, it appears that we will need a more extensive descriptor that can be submitted as a whole to the SCITT ledger. This is similar to the SBOM, but is for general data items.

1. For this use case, the data artifacts are sometimes very large, and there may be numerous data items that are logically part of one concept. Ballot images for one election are one concept, but may be many individual ZIP files, each containing say 50,000 image files. These individual files may be logically grouped into batches, which are individually secured, probably using a separate file which contains only certificates of some kind that were likely only one-sided certificates, without any endorsement by a SCITT Ledger. Furthermore, each image may have its own hash value, to allow any raster image (which may be embedded in a larger file wrapper, like PDF) to be compared and tracked from the source. For example, a PDF file containing an image will produce a different hash value if the file is copied to a new file, even if the image itself is unchanged. An image may be compressed using different methods and still be the same exact raster image when uncompressed. Tracking the image to the source may not be possible unless the hash of the core image, without compression and wrapping is available.

2. It would be convenient to combine all files and hash values for each one into say a DBOM JSON descriptor, which would then be the artifact which is submitted to a repository, and the hash of that would be included in the SCITT Ledger. It seems that this descriptor, which points to a set of large archives and other files that comprise the entirety of the artifact, would be small enough to be submitted to the SCITT ledger as a payload. Yet in general, separating the data payload from the minimal ledger entry is probably a wise choice, even if they may be closely associated.

3. The DBOM can deal with the intricacies of how a large number of individual image files, in this case, or perhaps in general any big data resources, so it can be cryptographically secured and thereby improve trust, and allow auditing of any vertical use case that is data oriented.

4. The concept of DBOM here is more than just metadata about the data, it includes (links to) the data itself. When compared to the software use case, this would be the SBOM

plus the code product itself.

5. Data conversion pipelines may have a number of intermediate steps, and it may be necessary for auditing purposes to maintain those intermediate data representations which are both required as inputs and the output produced by each stage.

6. There are already standards which have been sponsored by NIST for election data reporting that may have many of the fields needed for this specific case, and will be an appropriate starting point.

The following document considers the option for packaging arbitrary data.
https://docs.google.com/document/d/1xfU_s1Eu51z_WGg5VYBsQtjsKcrV6_TvFXj2WxBcj90/edit?usp=sharing

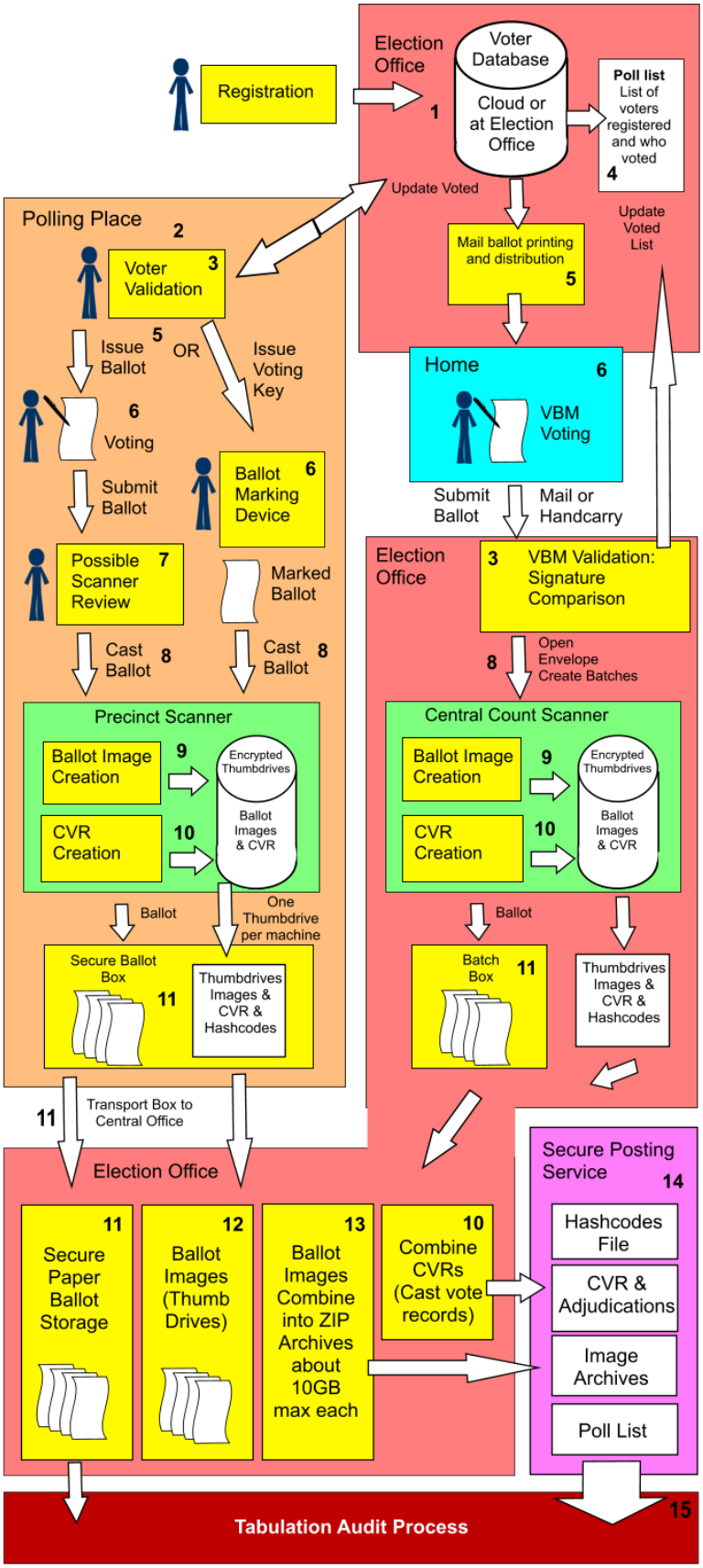
Conclusion

It appears now that there is nothing particularly special about the election data use case for the SCITT ledger and should not pose any difficult hurdles. But it does mean that a DBOM type descriptor will need to be developed that is general enough to be useful for this use case. This DBOM descriptor, which describes the full set of data that comprises the result of an election, will be the item submitted to the SCITT ledger, and it will be relatively small in size (probably <1MB).

Appendix 1: Election System Hazards

See the adjacent flow diagram. This first part does not include risks DURING auditing.

1. Registration:
    a. Malicious purges/ changes
    b. Early deadlines
    c. Registration in multiple jurisdictions
    d. Mitigation: same day reg./ERIC (election registration information center, see https://ericstates.org)
2. Polling Location Access
    a. Last minute changes, bogus announcements of location or date changes
    b. too many voters, long lines, inadequate machines
3. Voter Validation including Signature Comparison
    a. Difficult to obtain ID req. or inadequate identification requirements
    b. Signature comparison faults
4. Poll list errors
    a. Did the voter already vote?
5. Issuing ballot errors
    a. Req. to request correct ballot
    b. Correct ballot availability
    c. VBM ballots sent to wrong location
    d. Duplicate voting; deceased voter ballots used by others
    e. Ballot harvesting
6. Marking errors (voter Intent) & Submission problems
    a. Use of unreadable barcodes
    b. Confusing ballot design
    c. Incorrect barcodes printed on BMD ballot cards
7. Scanner Review Misdirection
    a. voter-facing system may misinform the user
8. Security of Cast Ballot prior to scanning.
    a. Higher risk in central scan operations
9. Ballot Image Manipulation prior to creating the CVR
    a. Higher risk in COTS scanner
    b. Limit risk by sampling paper
10. CVR Modification / Mismatch
    a. Voter Intent Misinterpretation
    b. Malicious CVR changes
11. Paper Ballot Security After scanning
12. Ballot Image on thumb drives (Modification)
    a. Would require matching changes to CVR
13. Ballot Image Archive Creation and posting
    a. Would require matching changes to CVR



More Information: CitizensOversight.org

Election Hazards

1. Registration:
   > Malicious purges/ changes
   > Early deadlines
   Mitigation: same day reg./ERIC

2. Polling Location Access
   > Last minute changes
   > too many voters, long lines

3. Voter Validation including Signature Comparison
   > Difficult to obtain ID req.
   > Signature comparison faults

4. Poll list errors

5. Issuing ballot errors
   > Req. to request correct ballot
   > Correct ballot availability

6. Marking errors (voter Intent) & Submission problems
   > Use of unreadable bar codes
   > Confusing ballot design

7. Scanner Review Misdirection

8. Security of Cast Ballot prior to scanning.
   > Higher risk in central count

9. Ballot Image Manipulation prior to creating the CVR
   > Higher risk in COTS scanner
   > Limit risk by sampling paper

10. CVR Modification / Mismatch
    > Voter Intent Misinterpretation
    > Malicious CVR changes

11. Paper Ballot Security After scanning.

12. Ballot Image Modification on thumb drives.
    > Would require matching changes to CVR
    > Thumbdrive tracking errors.

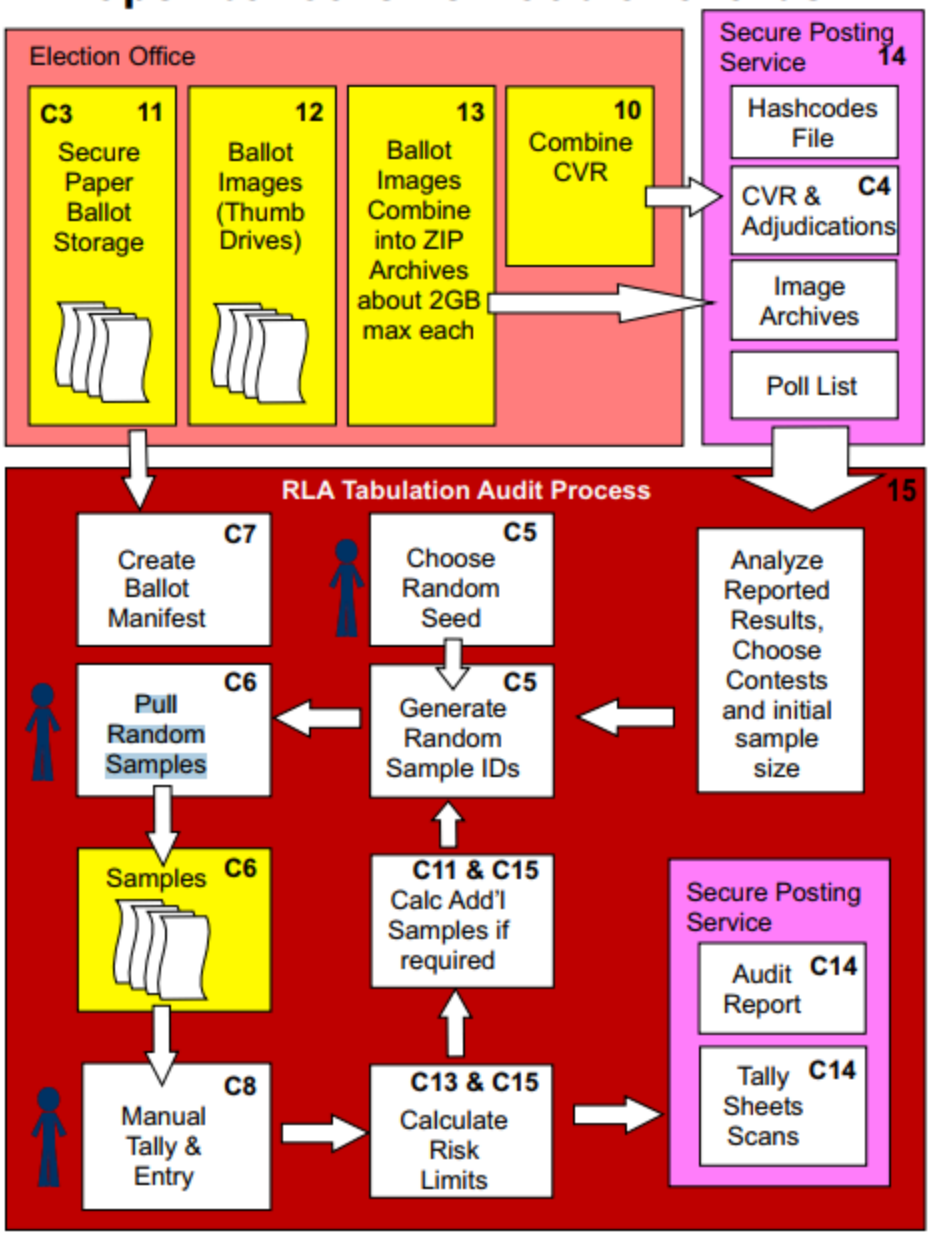
13. Ballot Image Archive Creation and posting
    > Would require matching changes to CVR

14. Posting Service Security
    > Posting service security is key to their business model.

15. Hazards within the audit process that can defeat the audit, including reporting.
    > Statistical audits that sample individual ballots have many hazards that may defeat the results.
    > Ballot Image Audits can be redundantly checked to reduce hazard of compromised auditor.

14. Posting Service Security
    a. Posting service security is key to their business model.
15. Hazards within the audit process that can defeat the audit, including reporting.
    a. Statistical audits that sample individual ballots have many hazards that may defeat the results.
    b. Ballot Image Audits can be redundantly checked to reduce hazard of compromised auditor

Paper-ballot review audit hazards



Paper-ballot review audit hazards -- Can defeat the audit if:

C1. Ballots are modified, added, or deleted prior to scanning.
C3. Ballots are modified, added, or deleted after scanning but prior to sampling.
C4. Cast-vote-records modified.
C5. Random sample generated to avoid hacked samples
C6. Drawing samples with to avoid hacked samples or in favor of desired option.
C7. Ballot manifest manipulation to avoid hacked samples or in favor of desired option.
C8. Manual Tally and data entry, "innocent fix-up", or DRE-like data entry w/o paper trail.
C11. May result in a full manual count if margins are close.
C12. May confirm a hacked election due to sampling error allowances.
C13. May not include all contests, esp. small contests.
C14. Incomplete or inaccurate reporting.
C15. Calculation mistakes or hacks.

More Information: CitizensOversight.org