Securing Digital Ballot Images to Enable Election Audits

Ray Lutz, CitizensOversight.org
raylutz@citizensoversight.org

2019-11-11 V0.1
2019-11-23 V0.2 (move to Google Docs, minor revisions)
2019-12-03 V0.3 -- Added section 7.1 on image file digest of core image only.
2019-12-06 V0.4 -- Adopted editing changes and comments by Maurice Turner and Lynn Garland
2019-12-09 V0.5 -- Improved definition of using hash values of uncompressed core image.
2020-06-06 - Added note regarding "remote attestation" and standards promoted by the "Trusted Computing Group"
2022-08-02 - minor edits to reflect support of FIPS 140-2 by VVSG 2.0

Web-based version of this document is here: https://copswiki.org/Common/M1936

1. Introduction

The intent of this document is to express some ideas regarding the security of digital ballot images to enable ballot image auditing, and reduce the possibility that the ballot images have been modified through a malicious attack. The ideas here include both legal liability of the device manufacturer and technology using Public-key Infrastructure (PKI) technologies, including cryptographic hashes, digital signatures, and other techniques. It is assumed that the reader is already familiar with PKI concepts[1]. Also, this is presented for further discussion among experts in the various fields of expertise.

By "Ballot Image" we mean a relatively high-resolution full-page digital image of a hand-marked (or machine marked) paper ballot, and not the (now deprecated) meaning of a digital record of a DRE (direct-recording electronic) equipment session with a voter, normally also called "touch screen voting." Unfortunately, many laws and guidelines still use this deprecated usage of the term, so any reading of other documents must be done with care when that term is encountered.

These ballot images can be used to audit the election by performing an exhaustive review of the ballot image to create an independent tabulation by an independent third party[2][3]. But such auditing is only any good if the ballot images are an authentic reproduction of the ballots themselves.

[1] https://en.wikipedia.org/wiki/Public_key_infrastructure
[2] https://auditengine.org
[3] https://clearballot.com/products/clear-audit

This document reviews hazards and challenges and proposes the creation of "Trusted System" for creation of the images, and proposes a scheme and protocol that enables verification of the code used to generate the images, review of the fidelity of the ballot images, so as to create a strong foundation for the creation of ballot image auditing services. These proposals are based on both a technological and legal framework.

2. Three Scanning Architectures

Voting equipment that scans hand-marked paper ballots to produce full-ballot images falls into three architectures. These systems are all used in conjunction with the Election Management System (EMS) software of the vendor of the equipment, which runs on a computer system in the central office. All the systems can provide images after being processed by the EMS so they can be archived or be posted to a publicly available web site.

2.1 Embedded scanner with Voter Feedback

All of the leading vendors -- Election Systems & Software (ES&S), Dominion, and Hart -- provide a system that can be deployed to voting locations, and that use an embedded scanner with an associated embedded computer system. All vendors can produce relatively high-resolution full-ballot images of at least 200x200 pixels per inch resolution.

Prior to being used, each one of the precinct scanners must be mated with a USB Drive (or equivalent) which includes configuration information for the election and precinct where the scanner will be used.

Election law provides that such a scanner must be able to provide feedback to the voter if contests are over-voted[4] or if the ballot is completely blank (which would likely mean that the voter is using some manner of marking that is not visible to the scanner.) The devices

ES&S DS-200

Dominion ImageCast Precinct

Hart Intercivic Verity Scanner

generally do not provide notices if the voters decided not to vote on some contests or if they under-voted contests, but generally configuration settings do allow it. The voter can back out and spoil that ballot and then hand-mark a new ballot if they want to, or they can go ahead and cast the ballot as marked.

These systems do not provide any feedback to the voter so they can inspect the quality of the ballot image itself.

These systems may have an associated "ballot marking device" (BMD) for disabled voters, as required by HAVA election law. The ES&S ExpressVote BMD system produces smaller ballot summary card with the votes encoded into barcodes and associated human-readable text listing the ballot selections. These are fed into the DS200 scanner and the vote is extracted by interpreting only the barcodes and not the human readable text. Hart and Dominion uses full-size ballots that are printed and marked by their BMD and are largely indistinguishable from hand-marked ballots. Dominion goes the extra mile by marking the ballot using a library of hand-marking scribbles that look like human marking. Some ballot marking devices actually do the tabulation on the spot rather than relying on re-scanning the ballots. This manner of operation we do not recommend. For the purposes of this document, we will focus on the scanners that are used to scan both hand-marked paper ballots and BMD marked ballots.

Internally, these scanners will not only create a raw scan, but also review that scan to determine the ballot style, which then provides information about where the targets (ovals or rectangles to be marked) are located and what contest and option they correspond to, and then to detect those marks and provide feedback to the user, and then to extract the vote and record it. They keep a running tabulation of the ballots typically by appending the interpretation of each ballot to a single file. That file and the ballot images are written to removable media, usually one primary USB Flash Drive (USB Drive) and also to a backup device. (Although these devices may use other forms of removable storage, such as SD-cards, we will call them USB Drives without implying any restriction in terms of form.)

It is now mandated[5] that the newer generation voting machine and scanners also digitally sign the cast-vote records and ballot images using FIPS 140-2[6] functionality, and typically also encrypts the data on the USB drive and also any data that may be transmitted by cellular modem.

---

[5] https://www.eac.gov/sites/default/files/TestingCertification/Voluntary_Voting_System_Guidelines_Version_2_0.pdf VVSG 2.0 see page 15 "Requires Federal Information Processing Standard (FIPS) 140-2 [NIST01] validated cryptographic modules (except for end-to-end cryptographic functions); Requires cryptographic protection of various election artifacts; Requires digitally signed cast vote records and ballot images"

[6] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf -- "SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES"

At the end of election day, the USB Drive, now containing the tabulated votes, the ballot images from that machine and the respective digital signatures is returned to the central office along with the ballots. The machines generally also have the capability of sending the result of the tabulation (totals for each contest) to the office using either wired telephone or using cell-phone connections[7]. Although these machines are not internet connected, the communication protocols on the wired telephone or cell-phone channels should still be secured using the best security available, such as TLS 1.3[8]. Due to the large amount of data, the ballot images are not transmitted in this manner. Also, due to the small chance that the communication might be intercepted using a MITM (Man in the Middle) attack[9], the data sent over such channels should be considered preliminary, with the data from the USB Drives used as the trusted data.

Also, these precinct scanner systems provide a storage compartment for ballots to be accepted in case the scanner system fails during the election, so they can be scanned later. This fact also underlines the requirement that initially transmitted data must be considered preliminary.

To maintain voter privacy, scanned ballots are not physically kept in order as they fall into a bin, and the ballot images are shuffled in memory[10], and typically are assigned a single timestamp to remove the possibility that the order could be reconstructed by using timestamps.

Since these devices are not internet-connected, the USB drive is the best and only means to configure the device and get both the ballot image data and ballot-by-ballot cast vote record (CVR). Therefore, the typical protocols of popular security schemes (such as the current state-of-the-art, TLS 1.3) are not available

We will note that these systems do some modification to the ballot images. At a minimum, they detect the direction and side (front/rear) the ballots were inserted into the scanner and will then flip the images so they are right-side-up, and ordered front and rear. They also must perform alignment and registration, and verify that the ballot can be read appropriately. Some go even further. For example, the Dominion scanner offers the "AuditMark" technology in their

---

[7] Many states do not allow any wired modem or cell phone connection to transmit preliminary results.
[8] https://www.ietf.org/blog/tls13/ TLS 1.3 updates the most important security protocol on the Internet, delivering superior privacy, security, and performance.
[9] https://en.wikipedia.org/wiki/Man-in-the-middle_attack
[10] For example, in ES&S DS-200 scanners, they set up 4096 folders and then based on a random number, assign each pair of image files (which are in zipped PBM format) to one of the folders. Even though this may look shuffled to a user who looks at the folder tree, the files are still likely written to flash memory in order and a detailed review of the contents of the USB drive could still reveal the order the files were written. Once the files are read from the USB drive and copied to another device, discerning the order in this manner would not be possible. This underlines the approach of tracking images by core hash rather than accessing the images directly from the USB drives.

ImageCast system, which adds raster data to the end of the ballot which provides human-readable text summarizing the vote as it was extracted from the ballot.

However, currently deployed systems do not provide a means to thoroughly inspect the quality of the ballot images for fidelity and clarity, or other problems. This issue will be discussed further in the section Image Fidelity.

2.2 Central count scanner using a custom-built embedded computer and document scanner

ES&S is the only major election equipment supplier that provides central-count scanners that are custom built and do not rely on an associated PC for processing the images, the DS-850 and DS-450 products. These systems include removable media but commonly are networked to the non-internet connected LAN, so they can send ballot images and CVRs as they go. They can operate very quickly (DS-850 can process 300 double-sided 14-inch ballots per minute) as they accept a stack of ballots and restack on the output. These scanners do keep the ballots in order so the CVR can be associated with the ballot scanned, and so it is feasible to use a ballot-comparison Risk Limiting Audit (RLA).

We must note that central scanning operations of this architecture and also that of Section 2.3 provide the possibility of utilizing systematic image quality control procedures that are used in trusted systems to produce authentic digital copies of paper documents. Ballot images are reviewed as they are produced to avoid poor digital images of those documents. Procedures include random sampling of images produced and evaluate the correctness and fidelity of the

images using statistical control by characteristics which are evaluated in the digital images. AIIM TR-34 "Sampling Procedures for Inspection by Attributes of Images in Electronic Image Management (EIM) and Micrographic Systems" (1996).

2.3 Central count scanner using a commercial off-the-shelf (COTS) scanner and an associated computer system

The other leading election equipment vendors, Dominion and Hart, offer central-count ballot scanning systems that use Commercial off-the-shelf (COTS) scanners that are not custom designed for the ballot scanning application, and rely on an associated computer (PC) to acquire and process the images. Since these systems are used in a central-count situation, there is no need to provide the option to voters to correct their ballot. Therefore, these systems can operate at much higher speeds, and are appropriate for districts that have a large number of Vote-by-mail ballots (VBM, also known as absentee ballots.)

Dominion ImageCast Central with COTS scanner

COTS scanners typically rely heavily on processing within the associated computer system but may also perform internal image processing. They do not internally understand the election application and can only produce raw scan data. Because COTS scanners are used in a plethora of applications, they are heavily tested by a much larger installed base of users.

3. Trusted System Concept

The electronic representation of documents is a trend that has been occurring for more than three decades. After microfiche was invented in 1961, images of documents were preserved on microfilm or microfiche to save space and avoid paper degradation. During the 1990s and 2000s, large warehouses of paper documents (and microfiche) were converted to electronic document images, so they can be quickly accessed and shared by more than one worker at a time.

Seeing the advantages of moving to electronic representation of documents, government regulations were modified to encourage the use of digital document images instead of physical documents. They addressed the ability to read documents without needing any additional information, and recommended the use of PDF/A-2 as the format used for data archival[11]. [Note that PDF files, if the same file is copied, will not provide the same hash value. TIFF does not have this problem as two identical TIFF files will produce the same hash value.]

But how can we trust that an image is an accurate representation of a document? This problem will always boil down to the concept of a "trusted system" and set of procedures which creates the document images.

In 2012, California adopted regulations that require state agencies to employ a trusted system for maintaining all electronic records created or stored as an official record. The State of California defines a trusted system as, “a combination of techniques, policies, and procedures for which there is no plausible scenario in which a document retrieved from or reproduced by the system could differ substantially from the document that is originally stored.” (*Source: California Government Code 12168.7(c)*)[12]

The CA SOS Electronic Records Guidebook[13] explains that agencies that wish to destroy paper documents and rely solely on electronic versions will need a trusted system in place. A trusted system certifies that electronically stored information (ESI) is an authentic copy of the original document or information.

A trusted system must include an avenue for maintaining at least two separate copies of an electronic resource. A combination of proper hardware and media storage techniques are necessary to prevent any unauthorized additions, modifications, or deletions to a document. A trusted system must also stand up to the rigors of an independent audit process that ensures that no plausible scenario for altering documents is feasible. Lastly, a trusted system requires that at least one copy of a stored electronic document or record is written that does not permit any unauthorized alterations or deletions and is stored and preserved in a separate and safe location.

---

[11] https://www.iso.org/standard/50655.html -- ISO 19005-2:2011 Document management — Electronic document file format for long-term preservation — Part 2: Use of ISO 32000-1 (PDF/A-2)

[12] "Trusted Systems" in the CA SOS Electronic Records Guidebook https://www.sos.ca.gov/archives/records-management-and-appraisal/electronic-records/electronic-records-guidebook/trusted-systems/

[13] https://www.sos.ca.gov/archives/records-management-and-appraisal/electronic-records/electronic-records-guidebook/

4. Attack Scenarios

In this section, we consider the range of possible ways that an attacker may modify the ballot images as they are created by the scanner and prior to being secured by the subsequent processes. We will primarily consider the embedded scanner system as described in Section 2.1 or 2.2. For the COTS scanner architecture of Section 2.3, modifications to the scanner device driver and PC-based software system are comparable with modifications to the embedded system, and attacks have been demonstrated[14]. We will specifically consider the following attacks:

1. **Firmware Substitution:** An attacker may open the scanner device and substitute firmware which has additional functionality to modify ballot images. Since the precinct scanner and custom central-count architectures have the capability of parsing the image enough to recognize voter intent, they also have the information required to "flip" a vote from one candidate to the other, such as knowing what location on the ballot should be modified. What is definitely normally missing is the additional capability to modify the image without leaving any evidence of alteration in the image. If that capability is added to the firmware, or is normally concealed and is then activated in the code, this would accomplish this attack.
2. **Pre-existing Backdoor:** An attacker may insert a USB drive which contains the additional code and rely on a pre-existing "backdoor" in the system where the code in the USB Drive would be used in conjunction with embedded code, and perform the alteration as described above. In HP laser printers, for example, the font cartridges normally had the word FONT as the first four bytes of the cartridge. Engineers discovered that the word CODE would cause the printer to attempt to execute code from the cartridge, and they also were able to reverse-engineer the rest of the printer system to be able to make use of its capabilities, and as a result Postscript interpreter cartridges were developed as unauthorized third-party aftermarket accessories. A similar back door could be constructed in election ballot scanners. But there may be other mechanisms that could be similarly deployed, particularly if there are any ports on the device.
3. **Firmware Modification Introducing a Backdoor:** An attacker may combine the two steps above, where a scanner that has no "backdoor" designed-in by the vendor, could be subverted by adding just a very small alteration to the base embedded code which

---

14 https://docs.google.com/document/d/1NudWuO7pD2vU_zQZaYq9-eus6ugaEHqLa688bnZF444/edit?usp=sharing "Critical Review of the paper 'Unclear Ballot'"

would then allow code in a USB drive to be activated and executed to perform the image alteration function.

4. **USB Drive Interception and Image and Tabulation Modification:** The USB drive could be intercepted enroute from the precinct to the election office. and the images and associated CVR could be modified before the data is acquired (i.e. read from the USB drives) by the EMS (election management system).
5. **Modification of Ballot Images and CVR after Being Received by the EMS.** The EMS could modify the ballot images and CVR after they have been transferred. There are many steps in processing after the ballot image data is acquired, including possibly changing the image format from a proprietary format to PDF/A and regrouping the files into precincts, etc. It is not therefore a simple matter to ensure that no ballot image data is modified. If the ballot images were modified, then the CVR would also need to be modified by the attacker. Otherwise, comparing the ballot images to the official CVR in a ballot-image audit would discover substantial differences.

All of these attacks will be reviewed after the protocol and legal framework is presented.

5. Strategy for Ballot Image Security

The proposed strategy with election systems is based on a combination of both technological procedures such as encryption and authentication, along with legal responsibilities of the election system vendor and the certification process of those systems. This document will also suggest a code validation technique that utilizes specialized embedded-system capabilities and the use of an active software escrow service.

The vendor and the certification process will create legal assurances that the system as designed can be trusted, and will create authentic digital ballot images of the ballots that are scanned, and there is no opportunity that the images can be modified without detection. This will be mated with a process where the system can be checked that it has not been modified by an attacker.

Of course, all systems may not function correctly, and even though images may be created and are not modified from the originals, they may not be an adequate copy of the original due to improper scan settings or defects in the scanner itself. In such cases, the paper ballots are still available for rescanning to produce images that are authentic copies of the originals.

6. Assurances by the Vendor

To create a trusted system, we must have a number of legal and technical assurances by the vendor, such as:

1. **No Image Modification:** The certified system has no code designed to modify images directly so they do not provide an authentic representation of the ballot that was scanned. If additional information is added to the image, as is the case in the Dominion AuditMark system, then the file formal shall allow the discrimination of the added data from the original data.
2. **No Backdoors:** The certified system has no "backdoors" to run code from an inserted USB Drive, either by executing the code directly or copying the code into the system to modify operation.
3. **Includes Verification Capability:** The scanner device and EMS code will be designed to implement the verification protocol as described below.

7. Verification Protocol

Since the embedded system does not have an internet connection with the EMS, it is not possible to carry out a complete secure handshake that includes the generation of one-time keys or anything of the sort. On the other hand, since there is no Internet channel to attack, the attack scenarios are vastly reduced.

There is a single one-way transfer of configuration data from the EMS to the deployed scanner, and a single one-way transfer of tabulation and ballot image data from the deployed scanner back to the EMS. These transfers are implemented as physical movement of the USB Drive, sometimes called "Sneakernet." The USB drive is secured physically during this transfer to deter attack, and it is difficult to impersonate the scanner without having its secret key.

The goal of the protocol will be to detect alterations in data or the code. If such an intrusion is detected, then the paper ballots shall be re-scanned.

(IMPORTANT NOTE:further review of this topic has revealed existing infrastructure within the concept of "remote attestation[15]," although it generally is envisioned to be used with connected "Internet of Things" devices that are directly connected, and does not include the notion of "Active Escrow Verification," which apparently is a new concept first described here. Nevertheless, most of this proposal is therefore somewhat moot as it would be preferable to use the existing extensive standards in this area, promoted by the Trusted Computing Group TCG,[16] for example)

1. **Private Key:** Each scanner deployed to polling locations shall include a private key that will be protected in hardware and infeasible to read from the device. The vendor will act as the certification authority. Because there is no internet connection, it is not

[15] https://web.cs.wpi.edu/~guttman/pubs/good_attest.pdf "Principles of Remote Attestation"
[16] "https://en.wikipedia.org/wiki/Trusted_Computing_Group" and https://trustedcomputinggroup.org/

possible to contact a trusted certification authority by the deployed device. The EMS has the public key of each scanner device. [This is generally the situation that is used today in scanners, but FIPS 140-2 capabilities are required in new voting systems that comply with VVSG 2.0, although no hardware security module is required.]

2. **Signed Configuration Data:** The EMS will write configuration information in each USB Drive and provide a signed hash that is not reproducible by any device except for the EMS system. The deployed scanner will not be able to authenticate this itself, but it can be checked later. (The goal here is to allow any scanner device to accept any USB Drive so it is possible to swap out failing scanner devices without undue logistics which would otherwise occur if each USB Drive were encoded with the specific device it was intended to be used with. However, it is currently the case that voting systems are required to create a single paired configuration device which will work on only the single voting machine it is paired with.)

3. **Salt from Random Beacon:** The EMS system will provide "salt" which will be used in the process of verifying that the code in the scanner is the same as the certified code. The salt has two components, 1) a timestamp of a time just prior to the election, and 2) a random number produced by a public random beacon based on that timestamp. A public random beacon protocol has been developed by NIST to provide 512 random bits that are created at periodic intervals[17]. The random value can be easily checked after the date it was created, but is unpredictable.

   Prior to election day, the EMS system shall access a random beacon service that will produce a random "seed" value based on the timestamp provided. It may be recommended that at least two independent random beacons are used to reduce the likelihood that one operator could be corrupted to produce predictable values. These values are placed in the USB Drive for each unit with other configuration data.

4. **Code Verification Signature:** When the scanner system is set up in the polling place, the USB Drive is inserted and the scanner is initialized. Upon initialization, the scanner must create a secure hash of the entire code memory space of the device, including all bootup code, combined with the "salt" random bit value passed. Because the salt value is provided by the random beacon immediately prior to the election, it is not possible for an attacker to pre-compute the proper hash value of the code. A verification signature is also produced using the secure key in the device. These are written into the USB Drive, plus a list of any unique values specific to this scanner device (should be minimized to the unique Device ID if possible) and the code space addresses where this device-specific data is stored. It is necessary to obtain this device specific data and

[17] The draft document is at this URL: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8213-draft.pdf

where it is located in the code so those values can be changed accordingly in the reference code.

5. **Images Redundantly Recorded and Signed:** The scanner device is used as usual to scan the ballots and store the ballot images and tabulation in the USB Drive and at least one duplicate memory device or internal device, shuffling the images in memory. Each image is separately signed with the private key in the device. [Shuffling is current practice at least by ES&S, according to communications from ES&S. However, most jurisdictions do not use redundant image storage, with the logic that they can always rescan the ballots if there is a failure.]
6. **Second Verification Signature:** When the polling location is to be closed, the scanner creates another secure hash of the code space and the data stored in the USB Drive, and then adds this to the USB Drive, and creates a signature block as well. This is not a standardized process and may not always exist.
7. **Chain of Custody Security:** The USB Drive is removed from the scanner device and placed in a physically secure container with a numbered seal and transported with paper ballots to the central election office.
8. **Verification Record Set:** Since the computer which hosts the EMS software is not on the Internet and it does not have a copy of the firmware of the scanner devices that it can compare with the hash values returned from the scanner, it is not feasible for the EMS software to validate the scanner software immediately. Instead, after all USB Drive data is acquired into the EMS system, a list of the validation values are combined into a file and that file is written to another USB Drive used only for validating the scanner software.
9. **Active Escrow Verification:** The verification data in that Validation USB Drive can be validated by sending the same random bit value to a validating escrow service, where the scanner firmware is securely archived. The escrow service must not only physically secure the code, but also be able to perform a validation service where it generates a secure hash digest of the reference code plus the salt and modified with the device-specific values also provided. The returned secure hash value can be compared with the value returned from the scanner device. If the hash of the code in the scanner device does not match the hash of the code in the escrow service, then the scanner device may have been hacked, and the ballots should be considered untrusted, and therefore rescanned.

This protocol will allow an honest vendor to detect a compromised scanner device. However, if the scanner device has a pre-existing backdoor, it could be compromised without any change in the primary code. The certification process of these devices must include careful

inspection of the code to determine that no backdoor exists, and the vendor must also provide an affidavit certifying that no back-door exists, and that if the hash codes provided in the process check out, that the code in the scanner cannot be compromised.

A similar process can be performed with respect to the EMS software, and additionally to validate the COTS scanner driver in the case of the architecture in Section 2.3.

7.1 Image Data Immutability

The scanner device should create hash values using a secure hash algorithm of the core image, or alternatively of files like TIFF or PNG which contain the image data. (However, PDF does not produce the same hash when the same file is copied, and thus it is difficult to check for that reason.) For example, ES&S uses the PBM (portable bitmap) file format in their initial file created by the scanner. There are two files for each side of a one-page ballot, with naming convention to describe which side (F or R) they are, and they are already rotated so the top of the page is at the top of the PBM file. This file is then compressed and stored as a zip file. Later, the front and back are converted to a PDF file. If we want to be able to verify that the image data is unchanged from the original, it will be necessary to extract the image data from each side in the PDF file, uncompress it, and apply the secure hash algorithm to create the hash digest of each side. These hashes can then be compared with those that were originally provided by the scanner, which should be signed by the scanner using its private key so it is infeasible to forge the image data.

7.2 Some Points

In embedded systems, raw image data or normalized image data should be returned.

If images are modified with additional scan lines to provide the Dominion AuditMark data, it should be easy to discriminate the original data.

The final data output by the EMS system should be PDF/A to comply with the standards set by the Trusted System standards, or preferably TIFF creates a stable file format that produces the same hash when copied to a new file and it can contain both sides of the image. PNG is also stable in terms of creating the same hash when it is copied, but it is not a multiple page format.

Scanning devices should have private keys that are infeasible to access. This is now commonly the case in systems that envision conducting secure communications in the IOT development space. If any new approach is devised, then it should be quickly adopted by

election scanners. It would be nice if the private keys were embedded in a Hardware Security Module[18] that would be infeasible to penetrate and obtain the private keys.

Scanner systems can sign the image data with that private key. Each image is separately hashed and signed. It is not necessary to encrypt the image data, but it is commonly the case when data is encoded on the USB Device.

The public key would be registered and the vendor would serve as the certification authority.

The scanner configuration should contain signed configuration data from the EMS system.
Configuration includes:

1. Random "salt" value which is generated by a random beacon at a given timestamp, which should be generated just prior to the election. At present, it appears necessary to have only one salt value for all USB Devices because each scanner has unique Device ID data encoded in the code space that would cause every validation hash to differ.
2. Ballot style data for the ballot styles supported by the scanner device and other election-specific configuration data.

The entire codebase in the scanner, including the boot segment and hard-coded Device ID is hashed with the random "salt" value and signed by device.

Scanner device returns:

1. Raw or normalized image data of each ballot.
2. Each ballot image is separately hashed and signed by device. Please note that the hash algorithm is applied to the uncompressed binary core image data as defined by the simple "Portable Bit Map" image standard[19], with any comments removed. This will allow the data to be checked later, regardless of the container the image is in, such as PDF, TIFF, PNG, etc. and whether the image is compressed.
3. The device returns
    1. the signed hash of the scanner code memory with "salt", and the timestamp to allow later confirmation of the random salt, and
    2. the values and location of any device-specific data in the code segment. Note, if there is no device-specific data in the code-segment, then an additional "salt" is required to insure that the returned validation hash will not be the same for all devices.

[18] https://en.wikipedia.org/wiki/Hardware_security_module

[19] http://netpbm.sourceforge.net/doc/pbm.html

EMS Software will:

1. Create a similar validation value for the EMS code and any COTS device drivers used.
2. Create a list of all validation hash values, the timestamp and random beacon "salt" and the locations and values of device specific data, for each device, and publish this data on a public posting service like Sharefile.com that provides unalterable trusted timestamps.
3. Vendor should, as soon as practicable, perform their own check of the validation values of the installed EMS software, driver, and firmware for each device by testing those values against reference code at a secure validating escrow service. [Note that the law governing current escrow services do not provide any active service to validate the code secured.]
4. The public can also perform the same validation of the software, drivers, and device code by processing the posted values using requests to the escrow service. [Again, this is a new concept.]
5. The Vendor shall certify in an affidavit that there are no "back doors" in their code, and if the firmware, drivers, and software validate against the reference code in the validating escrow service, that their devices will produce authentic ballot images that will match the paper ballots scanned.
6. Election officials, after acquiring all image data, shall upload it to a secure posting service, similar to Sharefile.com, which features unalterable trusted timestamps. With such trusted timestamps, it is infeasible to alter the image data once it is uploaded without also modifying the timestamps.
7. The ballot image data can then be submitted to a ballot-image auditing service, along with the official CVR and validating hashes. Produced images should be similarly signed by the EMS and certified that they are unaltered. PDF has ability to sign files and this may be a useful function.

8. Review of the Attack Scenarios

Regarding the attack scenarios identified above:

1. **Firmware Substitution:** This attack is defeated due to the mechanism of creating the validating hash of the entire codebase using the "salt" random value from the random beacon service, and then providing a mechanism for comparing with the escrowed reference code. Since these devices have limited storage, it would be obvious if the attacker attempted to preserve the original code so a comparable hash value could be

computed to impersonate the original system. (For the EMS system, this approach may be feasible however.)

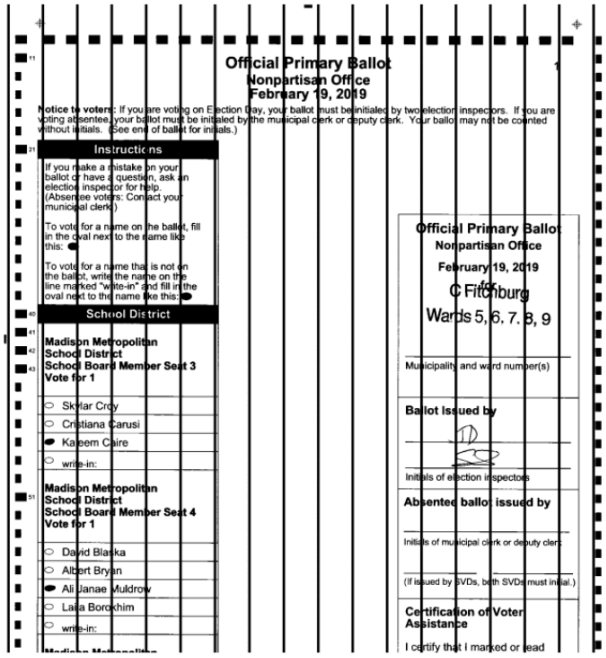
2. **Pre-existing Backdoor:** This attack is undetectable by the validation scheme and it relies on certification inspection of the code and the legal assurances of the vendor, which will make them liable if the code validates yet the images are found not to match the paper ballots in a random inspection audit.
3. **Firmware Modification Introducing a Backdoor:** Although this minimizes the changes to the code, it would still be detected by the validation scheme.
4. **USB Drive Interception and Image and Tabulation Modification:** First, the data in the USB Drive is signed using the private key in the scanner device and this is infeasible to obtain. But in the scenario where the attacker was able to someone obtain the same scanner and use it to produce another forged set of data, then first, it would raise suspicion if the data originally sent from the device right after the election (using a landline or cellphone connection) would differ from the data obtained from the device. But also, it would be necessary to produce the signed validation hash code. All these steps makes this attack infeasible in the short time, as it requires that the attacker have the same scanner, with its Device ID, embedded code and private key signing mechanism.
5. **Modification of Ballot and Images After Being Received by the EMS.**

   To subvert this attack outside the assurances of the vendor, it will be necessary to relate the ultimate image as provided in the PDF/A file back to the data provided in the USB Devices, which was provided and signed by the scanner device. This can be done even after the format of the image to PDF by converting the image back to the basic PBM format, creating the message digest hash value, and comparing the hash produced with the signed hash value that was originally provided by the scanner. Any amendments to the ballot images, such as is the case with the Dominion AuditMark scheme, should at least be removed to obtain the original authentic image. The original signed hash codes derived from the image data passed should be of the original image only, and should not include the AuditMark amendments.

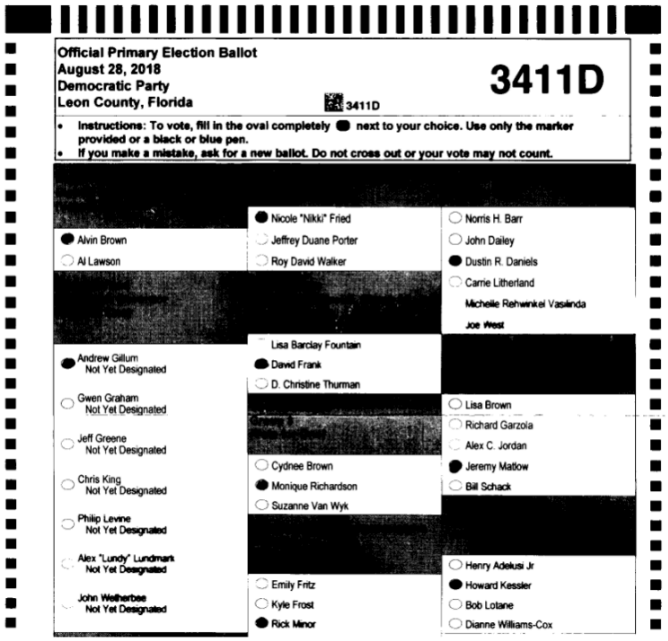
9. Image Fidelity Issues

Procedures that are used in trusted systems to produce authentic digital copies of paper documents include the review of ballot images as they are produced to avoid poor digital images of those documents. Although the central-scan architectures (2.2 and 2.3) can include review of ballot images to avoid image fidelity issues using inspection schemes such that

established by AIIM TR-34, in-precinct ballot scanners do not (today) provide any mechanism to review the ballot images as they are produced for fidelity.



Background stripes on ES&S Scanner



Threshold issues on Dominion Scanner

Once the ballots and the USB Drive has been returned to the central office, the ballots can still be checked for image quality, but such checks will not be able to include direct one-to-one comparisons of the ballots. Nevertheless, machine errors that might produce the vertical bars (shown above on the left), areas that are too dark (such as the image above on the right), or have spots on the image, can still be detected and in those cases, rescan the ballots using a central-count scanner. In some cases, such as thresholding issues that only impact supporting text, such as contest names, and not the areas where voters mark the ballots, these image defects may not impact the proper extraction of voter intent, but those defects can make it difficult for auditing software to perform OCR and to recognize the contests on the ballot, and to determine the ballot styles.

For the most part, the contest names are not required to confirm that the darkened oval corresponds with a given option, because the candidate names are generally quite distinct, particularly when viewed as a group. We can tell what contest it is without being able to read the contest names. However, in some ballot measures with yes/no ballot options, it will help to be able to read the contest names clearly in the image.

To avoid the thresholding problem, ballots should be designed without gray backgrounds as these do not scan well. Use black and white only, either black text on white background, or white text on black background. Both work well with OCR recognition.

The following Image fidelity attributes can be examined.

Attribute	Description	Comments
Image Size	Size of the image relative to the original, measured in both the horizontal and vertical directions	This attribute could be automatically checked by the scanner. Since the ballots are all of the same size in a given election, the image size can be automatically checked by the scanner and the ballot rescanned if the width or length of the ballot in the image is incorrect. Targets on the image can be used to check this attribute.
Image Skew	Rotation of the image or distortion where lines drawn on the page that are perpendicular are not in the image	This attribute could automatically be checked by the scanner, also by checking that targets in the image are in the expected locations.
Image Orientation	Ballots inserted top or bottom first will cause the image to be inverted.	Ballot scanners designed for scanning ballots will rotate the images so they are always in the correct orientation. COTS scanners do not necessarily do this, however.
Image Order	When a scanner images both the front and back at the same time (duplexing scanner), if the ballot is inserted top up or down, will result in the images in the other order.	Ballot scanners designed for scanning ballots will rotate the images so they are always in the correct orientation. COTS scanners do not necessarily do this, however.
Contrast	There should be a high contrast ratio between the text and the background.	Most ballots include known features, such as timing marks, that can be inspected to see if they provide adequate contrast.
Color Dropout	Specific colors are omitted from the image.	Some scanners, particularly of the optical scan type and not full ballot scanners, would print the ballot with the ovals of the targets in red, and the scanner would be designed to omit red in the image. This would make it easier for the simple reflective detectors to detect marks that were added to the ballot. In modern full-ballot scanners we do not

		recommend this feature and COTS scanners are never designed to drop out colors.
Poor Thresholding	Low contrast features are dropped out or a dark background obscures the foreground.	This is a common problem in ballot scanners, particularly when ballots are designed with gray background to emphasize the contest names. Although this is a recommended design by ballot designers to help voters understand the ballot, it is not recommended for use with bi-level scanners because the background frequently obscures the foreground.
Speckle or noise in the background	"White" areas of the ballots should not have any rogue pixels that are improperly black. The case of the vertical lines shown in the example would fill this case. Also, sometimes folds in VBM ballots cause dark lines to form across the ballot.	This is an important attribute for the ballot application because speckles in the area of a target may confuse the interpretation of voter intent. This attribute could be automatically checked by comparing the image with a known template and inspecting the areas outside the target areas.

If precinct scanners are provided with a screen with adequate resolution to display the ballot image the voter can inspect it prior to submission. The idea is not that the ballot would be inspected to see that it contains the exact vote by the voter, although that may be possible, but instead just to determine whether the ballots are not defective in terms of the fidelity of the image. Of course, there is no expectation that the voters will conduct quality control tests implied by the table above.

10. Conclusion

Improving the trustworthiness of the systems used to create ballot images is feasible, but it will require both technological improvements to allow validation of firmware and software by using a validating escrow service, and required mechanisms within the scanner device. Even with those in place, there is still the possibility of a back-door which may be undetectable through technical means, and thus the legal assurances are still required.

Since validating escrow services are not currently available and are not required by law, and the mechanism for validating firmware using embedded validation code does not exist in systems today, those nice features will not be available.

It seems likely somewhat easier to add the validation scanning mechanism to EMS software and driver validation compared with making changes to the installed base and changing embedded firmware in precinct scanners.

What is important is to provide for continuity of the ballot image data so it can be traced back to the original image provided by the trusted system, and to access the ballot image data as early as possible in the process.

It is clear that this subject will require discussion among the stakeholders and experts in the field. Clearly, the code validation infrastructure will not be available in the current generation of election equipment.