Scantegrity III: Automatic Trustworthy Receipts, Highlighting Over/Under Votes, and Full Voter Verifiability

Alan T. Sherman
UMBC CDL

Russell A. Fink
UMBC CDL

Richard Carback
UMBC CDL

David Chaum
Voting Systems Institute (VSI)

Abstract

Building on lessons learned from the November 2009 Scantegrity II election in Takoma Park, MD, we propose improvements to the Scantegrity II voting system that (1) automatically print trustworthy receipts for easier on-line verification, (2) highlight ballot features including over/under votes to comply with the *Help America Vote Act*, and (3) achieve full voter verifiability by eliminating print audits. We call the improved voting system Scantegrity III, which features a new ballot style and a special casting station that highlights ballots and prints receipts. Scantegrity III addresses the major limitations of Scantegrity II and delivers the feature most requested by voters and election officials at the Takoma Park election: printing receipts automatically.

We present, analyze, and compare three designs for a Scantegrity receipt printer: a simple image duplicator available to voters in an optional separate station before casting; a mark sense translator, connected to the official ballot scanner, which reads encrypted codenumbers printed on the ballot; and the Scantegrity III casting station, which is an embellished mark sense translator. At the Scantegrity III station, voters cast ballots that include both Scantegrity II codes in invisible ink and Scantegrity I codes in conventional ink; this combination of codes enables print audits to be eliminated. We also design a Trusted Platform Module (TPM) enhancement to bolster privacy, to store keys and verification codes, and to ensure that the correct software is booted. Election integrity does not depend on the correct operation of the TPM. Receipt printers reduce the amount of special voter instruction required, improve accessibility, enable each voter to detect if any additional mark is added to her ballot after casting, and make vote verification easier.

Keywords. Applied cryptography, End-to-End (E2E) election systems, print audit, Punchscan and Scantegrity voting systems, receipt printers, security engineering, Trusted Platform Module (TPM), trustworthy computing.

1 Introduction

The Scantegrity II voting system [14, 13] has significantly simplified the voter experience for End-to-End (E2E) voter-verifiable elections. On November 4, 2009, 1,723 voters elected the mayor of Takoma Park, MD, using this system [7]; Takoma Park will use Scantegrity again in their 2011 municipal election. Although the 2009 election was a success free of any major problems, this experience highlighted several limitations of Scantegrity: (1) some voters did not write down the exposed Scantegrity II codenumbers as required to verify their votes later on-line; (2) the print audit to check the correctness of the printed ballots added cost and complexity; and (3) the scanners used at Takoma Park were not compliant with the *Help America Vote Act (HAVA)* [1] because they did not notify the voter of over- or under-votes. In post-election surveys and informal interviews [8], both voters and election officials offered as their main suggestion that Scantegrity should include a way to print privacy-preserving receipts automatically.

Scantegrity provides *End-to-End (E2E)* voter verifiability. Each voter uses an optical scan paper ballot with special invisible printing in the markable positions. To select a candidate, the voter marks her ballot using a special pen with reactive ink that reveals a hidden confirmation number—called a *Scantegrity II code*—in each marked position. The voter can record the numbers revealed by her selections to check on an election website later. In doing so, the voter validates that her vote was recorded and tallied correctly, which helps verify election integrity without revealing how she voted. Each code is chosen randomly for each contest (race) and ballot, and therefore does not reveal the corresponding vote. The ballots are handled as traditional ballots, preserving the ability to count them by hand and to perform other election activities associated with optical scan systems.

Election integrity is verified when enough voters check their codes on the election website, and challenge any incorrect results. In 2009, some Takoma mayoral voters did not record exposed codenumbers. Some did not realize that they could make a receipt, and that doing so was necessary for on-line verification.[1] Some voters had trouble reading or writing codenumbers, and transcription errors always are possible when writing down the codes. Further, ballots with many races (or with many candidates when instant runoff is used) would burden the voter with recording multiple codes.

Securely adding a receipt printer to Scantegrity is a challenging task: for the system to preserve E2E voter verifiability, the voters must be able to verify that the printed receipts are valid. If a malicious receipt printer could generate improper receipts unnoticed by voters, changes in election outcomes might go undetected. Furthermore, a printer presents a potential vulnerability against voter privacy and adds system complexity.

Fink [18] and Carback [9] proposed a simple receipt printer with scanner, which in an optional separate step before ballot casting, duplicates the scanned images of the marked ovals, letting the voter compare her ballot and printed receipt without doing any character recognition. This design does not guarantee that the ballot cast is the same ballot scanned by the receipt printer. Fink [18] and Carback [9] also proposed a more complex mark-sense translator design in which the official precinct-count optical scan (PCOS) scanner is connected to a secure receipt printer. From the marked oval positions detected by the scanner, the receipt printer reconstructs the associated codenumbers independently, providing a powerful check on the PCOS scanner. Disadvantages include the need for the printer to know the codenumbers and the need for a physical mechanism securing the ballot against modifications before casting while the voter compares the printed receipt and ballot.

Embellishing the mark-sense translator, Chaum proposed Scantegrity III, addressing all of the main limitations of Scantegrity noted above and enabling safe automated receipt generation. In the Scantegrity III casting station, the voter places the marked ballot under a glass panel, where it is scanned and where ballot characteristics (including possible under- and over-voting) can be highlighted with backlighting. The device prints a receipt in one of two different styles verifiably chosen at random and permits the voter to compare the receipt and ballot before casting while they remain locked under the glass. Using a combination of ideas from Punchscan and Scantegrity I, this receipt construction eliminates the need for separate print audits. Significantly, the backlighting feature of this design is of separate interest and can perform more than simply identifying possible over- and under-votes.

Trustworthy receipt printers help increase the confidence of E2E election outcomes. They make it easier for voters to obtain receipts, and they facilitate the possibility of printing multiple copies of receipts. For example, an extra copy of every receipt could be printed and made available to independent auditors, who could check them and post them on-line.

Contributions of our work include:

- Three design concepts for a Scantegrity receipt printer, including a simple image duplicator, a mark-sense translator, and a Scantegrity III casting station.
- Solutions to three additional limitations of Scantegrity: eliminating the need for separate print audits, mitigating the threat of adding marks to ballots after casting (this vulnerability also exists for traditional optical scan systems), and notifying the voter of over- and under-votes prior to casting.
- A new user interface for any optical scan voting system featuring backlighting of ballots to highlight certain characteristics, including but not limited to over- and under-votes.
- Design enhancements for bolstering the receipt printers with a Trusted Platform Module (TPM), to safeguard privacy, detect problems sooner, and enforce election policy.

The rest of the paper is organized as follows. Section 2 reviews previous and related work. Section 3 presents our three receipt printer designs: image duplicator, mark sense translator, and the Scantegrity III casting station. Section 4 lists requirements for a basic receipt printer. Section 5 suggests how each design can be improved with a TPM to protect privacy and to detect problems sooner. Section 6 presents security arguments and discusses a variety of issues raised by our designs, and Section 7 concludes our work. Appendix A provides additional design and use details.

2 Previous and Related Work

Scantegrity II [14, 13] is an End-to-End (E2E) voting system, meaning that it provides high assurance that the tally is computed properly while maintaining ballot secrecy [33]. It has been deployed in a mock election [39, 40] and a real election at Takoma Park, MD [7, 8]. Printing the codenumbers in invisible ink simplifies dispute procedures of the earlier Scantegrity I system [12]. Responding to suggestions from Takoma Park voters and election officials to automate receipt printing, in a joint chapter of their dissertations, Fink [18] and Carback [9] describe two receipt printer designs for Scantegrity: an

[1] Some voters later explained that they chose not to read any instructions because they knew how to vote [8].

image duplicator and mark sense translator. Circa February 2011, Chaum drew preliminary sketches for a Scantegrity III casting station. This paper integrates and refines these ideas.

Aspects of our designs were inspired by Scantegrity's predecessors: Chaum's SureVote [12] has a receipt printer, and each voter in Chaum's Punchscan [32, 20] may take home as her receipt one of the sheets of the two-sheet ballot. In Punchscan, receipts preserve ballot privacy through indirection: the top ballot sheet reveals the permutation (cyclic shift) of (Scantegrity I-like) codenumbers for ballot choices per race on that ballot (but not the marked choice), and the bottom ballot sheet reveals the code of the marked choice (but not the permutation of code).

Neff's VoteHere system, in its instantiation with the Sentinel as envisioned by the Maryland Study [38, 29, 30], uses a receipt printer as an integral part of its protocol to commit cryptographic codes to the voter [28, 5, 4]. No element of SureVote or VoteHere guards against a malicious printer violating voter privacy.

Printers also appear as VVPAT devices attached to DREs. Studies have shown these systems to have serious usability, reliability, and security problems [21, 38].

Popoveniuc and Regenscheid [34] propose the Sigma ballot as a means to eliminate Scantegrity print audits. The main version of their system requires candidates to appear in random order per ballot, a feature that violates policies of many election precincts. Their system uses two specialized photocopy machines, which raise questions about ballot privacy and hence coercion.

E2E systems have their origins in 1981 work by Chaum [11], who first proposed cryptography for the purpose of anonymizing ballots in a verifiable manner. Adida [2] surveys the next two and a half decades of work in this area. The first proposals that can be identified as E2E were Chaum's SureVote and Neff's protocols mentioned above. In addition to Punchscan, other proposals include *Prêt à Voter* [16], the proposal of Kutylowski and Zagórski [26] as *Voting Ducks*, and Simple Verifiable Voting [6] as *Helios* [3] and *VoteBox* [37].

Fink and Sherman [17] describe the benefits *Trusted Platform Modules (TPMs)* can bring to voting in privacy and detecting problems sooner, even for E2E systems. They explain how end-to-end integrity does not guarantee end-to-end security, and how trustworthy computing (albeit imperfect) can meaningfully enhance election system security beyond outcome integrity.

Fink *et al.* [19] give a detailed design with protocols for using a TPM to reduce the trust base of a DRE and to enforce election policy; they also review prior work involving TPMs in voting. No previous approach uses a TPM to secure receipt printers.

For features and references on the TPM, see the TCG specifications [42] and the overview by Pearson *et al.* [31]. To developers using the TPM, we recommend the practical guide by Challener *et al.* [10] and the TrouSerS software stack and test suite for understanding programming details [23].

To the best of our knowledge, we are the first to propose backlighting paper ballots, though this application is related to the concept of semantic light explored by Lohr and Segall [27].

3 Three Receipt Printer Designs

We present three designs for a Scantegrity receipt printer: a simple stateless image duplicator, a stateful mark sense translator connected to the PCOS scanner, and the Scantegrity III scanner/printer station. Each design prints a paper receipt that the voter may take home; this receipt includes the ballot's on-line verification number and the codenumbers of the marked ovals. Our three designs offer differing engineering tradeoffs among simplicity, security, usability, and other features. Section 5 explains how to enhance these designs with a TPM.

Each of these designs assumes that the user has marked a paper optical-scan ballot prior to requesting a receipt. Consequently, each design enjoys a failsafe mode of operation that is simply Scantegrity II without any receipt printer, in case the receipt printer technology fails. Other design choices are also possible, including the possibility of adding ballot-marking capability to the Scantegrity III station. For any design, the voter may optionally create her own additional hand-written receipt.

Any design must address the following questions. (Q1) How does the printer know what codenumbers to print? (Q2) How does the voter check the validity of the receipt? (Q3) How does the system ensure that the ballot scanned by the receipt printer is the ballot cast at the PCOS scanner? (Q4) How does the receipt printer affect the voter's experience and flow through the voting process? Each of our designs reflects different answers to these questions.

There are three possible ways that a receipt printer could learn what codenumbers to print. The device could copy the images of the marked ovals without any optical character recognition (OCR); the device could perform OCR on the marked ovals; or the printer could determine the positions of the marked ovals and infer the corresponding codenumbers from privileged information. This privileged information could be entered into the device (*e.g.*, onto its TPM) or transmitted via the ballot in encrypted form using a key known by the device (*e.g.*, and stored on its TPM). We dislike the option of performing OCR since that would create significant reliability issues. Our image duplicator copies the marked oval images, and our other two designs infer the codenumbers from the marked positions.

To check the validity of a receipt in the polling place, the voter must compare the printer receipt with the marked ballot. Additional checks by voter or a designee may also be possible after leaving the polling place (*e.g.*, verify a digital signature, check codenumbers and commitments on-line). It is desirable for voters to check all digital signatures in the polling place (when they could immediately raise a complaint to a poll worker), though current policies and practices do not faciliate this goal.[2]

If the receipt printer is a separate optional station in the voting process, it becomes problematic to ensure that the ballot that generated the receipt is the same ballot without any modification that is later cast (and even if the ballot is the same, there is no guarantee that the PCOS and receipt printer scanners interpret the ballot identically). If the PCOS scanner and receipt printer are integrated as a single station, then a physical mechanism is required to lock the ballot where it can be seen but not modified, while the voter checks the receipt and before the ballot is cast. Our image duplicator exists as a separate optional station, and our other two designs integrate the receipt printer with the PCOS scanner in different ways.

We now describe our three designs. For each design, we summarize the voter experience; Appendix A provides additional design and use details.

3.1 Image Duplicator

The image duplicator is an integrated device comprising a scanner and printer. It scans the images of all markable positions and the ballot's two-dimensional barcode (qrcode). No other region of the ballot is scanned. As shown in Figure 1, the device prints the images of the markable positions, exactly as scanned, in a permuted order (by decreasing average pixel density) onto the receipt. It also prints the ballot's on-line verification number, which it read from the qrcode, and a digital signature. This design is stateless in the sense that it requires no knowledge of the codenumbers and no connections to the separate PCOS scanner. The device needs to know the locations of the markable positions, which can be communicated in the ballot's qrcode.

3.1.1 Voter Experience: Image Duplicator

A typical voter experience proceeds as follows, if the voter chooses to use this optional device.

1. The voter marks her Scantegrity ballot in the voting booth.

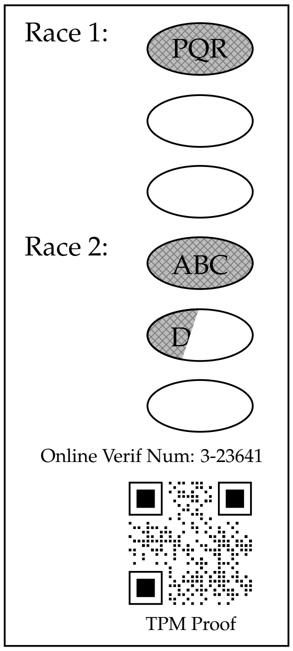

Figure 1: Receipt from image duplicator. For each race, images of scanned ovals are printed in order of average pixel density. Partial marks within ovals appear.

2. The voter presents her marked ballot to the image duplicator.
3. The image duplicator scans the ballot's qrcode (which includes the on-line verification number) and the ballot's markable positions (whether marked or not).
4. The device prints a receipt containing (a) for each race, images of the marked positions exactly as scanned, but with their order rearranged; (b) the on-line verification number; and (c) a signed digest of the on-line verification number, as explained in Section 5.
5. The voter compares the receipt with the marked ballot to verify that (a) the exposed codenumbers agree, and (b) the on-line verification numbers agree. If there is a discrepancy, or if the receipt is illegible, she alerts a poll worker who invalidates the ballot and logs the event, following the precinct's practices and procedures.[3]
6. The voter brings the marked ballot to the PCOS scanner, and she takes the receipt home.
7. After leaving the precinct (better: in the polling place if allowed), optionally the voter may, with the

[2] In the future, voters might be permitted to bring a trusted assistive device (*e.g.*, iPhone-like device without camera) into a polling place.

[3] For each design, if the voter so alerts a poll worker, the voter's marked ballot will be exposed to the poll worker.

help of a tool of her choice (*e.g.*, her iPhone), verify the digital signature on the receipt. If the signature is malformed, the voter may file a complaint.

8. Optionally, the voter (or anyone she so designates) may verify her vote on-line. To do so, she points a browser to the on-line verification website, enters her on-line verification number, and for each race compares the displayed codenumbers with the corresponding ones on her receipt. If any of the codenumbers do not match, she may file a complaint.

3.2 Mark Sense Translator

The mark sense translator connects directly to the PCOS scanner. It receives mark sensed positions from the PCOS scanner, translates the positions into Scantegrity II codenumbers (confirmation codes) that should be revealed on the ballot, and prints the codes onto a paper receipt. Unlike the image duplicator, the mark sense translator requires knowledge of the Scantegrity codes for each cast ballot, making it a stateful design. It also reports a count of the number of receipts printed, to support auditing, and provides a voter-verifiable check on the behavior of the PCOS scanner.

To enable the voter to check the printed receipt against the marked ballot, the PCOS scanner is equipped with a mechanism that holds the ballot under a glass panel after scanning and before casting. After checking the receipt, the voter either casts the ballot or ejects the ballot back to the voter.

Figure 2 shows how the voter interacts with the mark sense translator.

3.2.1 Voter Experience: Mark Sense Translator

A typical voter experience in the polling place proceeds as follows.

1. The voter completes her Scantegrity ballot in the polling booth, then presents her ballot to the PCOS scanner.

2. The PCOS scanner scans the marked ovals and qr-code (containing the on-line verification number) from the ballot.[4] The scanner interprets the marked ovals as selections (*e.g.*, "race 3, choice 2"), and sends them and the on-line verification number to the mark sense translator. Other data, such as overvote or undervote details compliant with HAVA requirements [1], may also be transmitted.

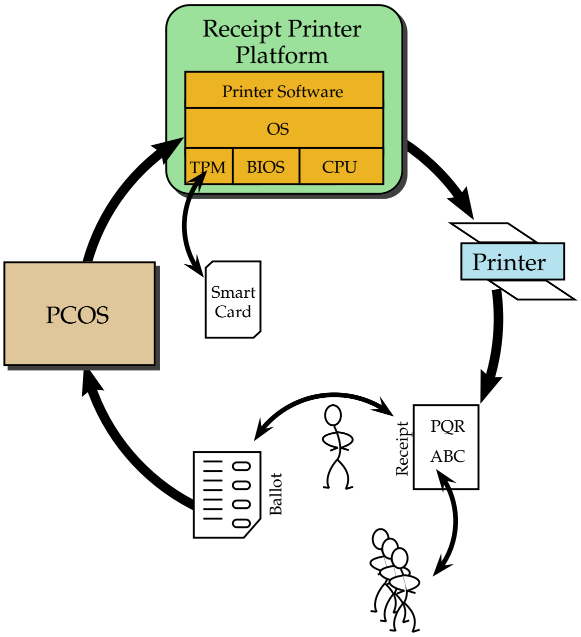

Figure 2: Voter interaction with the mark sense translator. The voter submits her ballot to the PCOS scanner, which sends the marked positions and on-line verification number to the mark sense translator. Optionally, the voter may verify the platform using a smart card and TPM. The receipt printer recovers the Scantegrity II codes, matches them to the marked positions, and prints the receipt. The voter validates the receipt against the ballot before casting.

3. The mark sense translator securely retrieves the Scantegrity verification codes for each selection (*e.g.*, as printed in encrypted form on the ballot), and prints them onto a paper receipt. It also prints a signed digest of the voter's codes, as explained in Section 5.

4. The voter verifies that the Scantegrity codes on the receipt, and the on-line verification number, match those on her ballot. She does so while the ballot is still locked underneath a glass panel. If the voter is satisfied, she may press a button to cast her vote and take her receipt home. If there is a discrepancy, she alerts a poll worker and her ballot is retrieved and revoked, consistent with the precinct's policies and procedures.

[4] In E2E voting, each voter must be able to identify on the public bulletin board the official data (*e.g.*, codenumbers) associated with her vote. Scantegrity enables this task with a unique on-line verification number printed on each ballot. But in some states, it is against election policy for ballots to have serial numbers or for the PCOS scanner to read ballot serial numbers.

3.3 Scantegrity III

In Scantegrity III, as with the mark sense translator, there is an integrated PCOS scanner/receipt printer casting station with one scanner, one printer, and a physical mechanism to lock the ballot and receipt under a glass pane after scanning and before casting. As does the mark sense translator, the casting station infers which Scantegrity II codes to print from the marked positions and from privileged information (*e.g.*, encrypted codes written in the qrcode on the ballot). Three innovations distinguish this new device.

1. The device can print receipts in two different types. For each voter, the device verifiably randomly chooses which style to print. As in Punchscan, together the two receipt types reveal the ballot choices, but taken alone, neither receipt reveals any of the voter's selections.

   A "verifiably random" choice is a random and unpredictable choice for which voters can verify that the correct process was followed. We assume that the device includes a simple, observable automatic physical mechanism to generate truly random bits (*e.g.*, using a die). These bits become part of the official election data posted on the bulletin board.

   For example, a fair die with red and green sides could be tumbled in a clear sealed dome with a simple camera sensing the outcome of each roll. The device prints receipt type 1 onto green paper, and receipt type 2 onto red paper. The voter verifies that the color of her receipt matches the color showing on the die.

2. Ballots contain additional codes: for each race, for each candidate, there is a Scantegrity I (S1) code printed in conventional ink. These S1 codes define a cyclic shift of the candidates, which are printed in a fixed order.[5]

   During election setup, the Election Authority (EA) publishes separate commitments binding each S1 code to its corresponding Scanterity II (S2) code, and binding each race to its S1 cyclic shift. As for Scantegrity II, we assume that the EA follows proper procedures to ensure all cryptographic commitments are correctly posted.

3. The device includes a usability feature of independent interest: when the ballot is placed under the glass panel, an LED display behind the ballot can illuminate aspects of the ballot of interest (*e.g.*, under- and over-votes). A second vertical LED display provides instructions to the voter.

The first two innovations combined eliminate the need for separate print audits, since the voter can later check on-line the unlocked commitments associated with the S1 codes revealed on the printed receipt, as explained below. The indirection permits the EA to unlock these commitments without revealing the voter's selections.

We now explain the Scantegrity III station in more detail by describing the ballot, the receipt types, a typical voter experience, and why Scantegrity III eliminates the need for print audits.

3.3.1 Scantegrity III Ballot

The ballot is a Scantegrity II ballot with added Scantegrity I codes printed in conventional ink. For each race, candidates are printed in a fixed order. The S1 codes define a cyclic shift of the candidates. To describe this cyclic shift, it is sufficient to specify the S1 code of the first candidate in the fixed order.

As with Scantegrity I, for privacy, it is important to protect the unmarked ballots so that, for exmple, the adversary cannot learn the mappings from on-line verification number to S1 codes. This mapping, together with the second receipt type, would reveal the selected candidates, which would violate voter privacy if the attacker could associate a voter with her on-line verification number. We assume that ballots are delivered securely to the precincts in tamper-evident sealed containers. Additional protection is possible by printing each on-line verification number in invisible ink or covering it with a scratch-off surface. Scantegrity II and traditional optical scan also have a variety of privacy vulnerabilities when the attacker has access to ballots before or after voting.

During election setup, the Election Authority (EA) cryptographically commits to the S1 and S2 codes, and to the binding between them. For each race, the EA publishes a separate commitment of the S1 code associated with the first candidate. It also publishes a commitment of the S2 code associated with each S1 code. Publishing these commitments does not reveal the associations.

Adding this layer of indirection between the candidate and its associated S2 code permits the voter to verify the correctness of both her ballot and receipt without revealing how she voted.

3.3.2 Scantegrity III Receipt Types

As shown in Figure 3, there are two types of receipts, both of which reveal the S2 codes of the marked choices. Inspired by Punchscan, receipt type 1 gives the cyclic shift of the candidates without revealing the marked

[5] S1 codes are letters written in alphabetical order, one per candidate. A *cyclic shift* specifies a rotation of these codes relative to the fixed candidate list, allowing the first candidate's code to be any letter and not necessarily 'A'.

choice. Conversely, receipt type 2 gives the S1 code of the marked choice without revealing the cyclic shift.

In receipt type 1, to specify the cyclic shift, it is sufficient to print the S1 code of only the first candidate in the fixed order. Doing so helps keep the receipts short, even if there are many candidates. The length of each receipt is linear in the number of contests, independent of the number of candidates.

At the casting station, the voter may compare her receipt and ballot for consistency. Later, during the optional on-line verification, the voter may further check her receipt against certain commitments which are unlocked after polls close. Specifically, if receipt type 1 is printed, the system unlocks its commentment of the S1 code for the first candidate. Similarly, if receipt type 2 is printed, the system unlocks the commitment between the S1 code of the marked choice and its corresponding S2 code. The choice of which receipt to print is determined in a verifiably random way.

If the voter does not wish to verify her vote on-line, she may insert the ballot and cast it without checking the receipt.

3.3.3 Voter Experience: Scantegrity III

1. The voter marks her ballot in a voting booth and then proceeds to a casting station. We shall assume that the ballot is not more than one page.

2. In the casting station, a vertical LED screen displays a few-second instructional video loop. It shows placing a ballot on top of a horizontal LED screen, face up, and closing the clear cover over the ballot. The horizontal LED screen under the cover displays supporting graphics during this process, such as multilingual text: "place ballot here."

3. Once the cover is closed over the ballot, an electrically operated latch secures the clear cover so that it cannot be opened. The ballot is scanned, perhaps using a video camera, and the device attempts to recognize the marked positions and the qrcode. If the scan is unsuccessful, the device unlocks the cover and asks the voter to re-adjust the ballot, returning to a variant of Step 2.

4. Three things now happen in parallel:

   (4a) The LED screen under the ballot draws attention to certain characteristics of the ballot. For example, marked positions might be highlighted by backlighting in one color. Similarly, over-votes can be highlighted with a distinctive color, perhaps with blinking; and under-votes can be highlighted with another color.

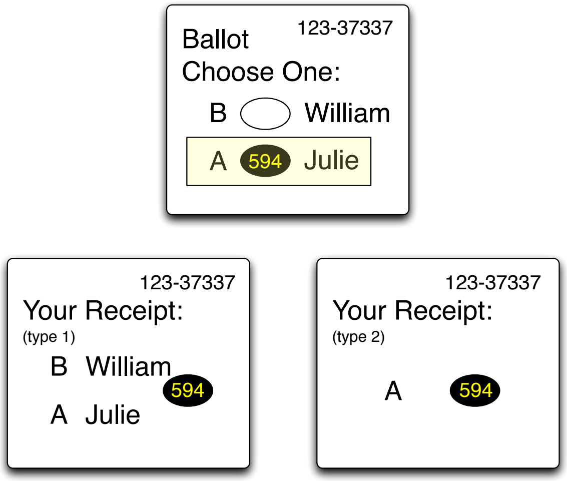

Figure 3: A marked Scantegrity III ballot with the two receipt types. For each race and choice, the ballot has both a Scantegrity II codenumber (printed in invisible ink) and a Scantegrity I codenumber (printed in conventional ink). Both receipt types include the Scantegrity II codes of the marked choices. Receipt type 1 gives the Scantegrity I codes (cyclic shifts) of the candidates, but does not reveal the voter's selection. Receipt type 2 gives the Scantegrity I code of the marked choice, but not the position (cyclic shift) of the marked choice.

   (4b) A receipt is printed and appears behind a second clear door, with a backlight that is illuminated. The system decides which of the two types of receipts to print by a verifiably random choice. The voter now may compare the ballot and receipt, checking the consistency of the on-line verification numbers, S2 codes, and S1 codes. If the voter discovers an inconsistency, she may summon a poll worker and file a complaint.

   (4c) The vertical LED asks the voter if she would like to cast her ballot, and if so, to open the clear cover and take the receipt. If the voter does not wish to cast her ballot, she is instructed to pull the cancel lever and take the ballot back.

5. If the voter opens the receipt door, the ballot drops into the ballot box. The voter removes the receipt which she may bring home. If the voter operates the cancel lever, the receipt drops into a receipt box. In either case the device returns to Step 1.

3.3.4 Eliminating Print Audits

Election integrity depends on voters being able to detect incorrectly printed ballots. A crucial aspect of ballot correctness is that the correct codenumber appears next to each candidate. Traditional Scantegrity II depends on a randomized checking process that yields strong statistical evidence that the voted ballots are correctly printed, but the cast ballots are not audited.

In traditional Scantegrity II, any voter may "print audit" her ballot [14, 7]. During a print audit, a poll worker marks the ballot as print audited, and the voter exposes all of the codenumbers. The voter may then copy the exposed codenumbers by hand or photocopy the ballot. Later, the voter may check that the codenumbers exposed in the print audit match those listed on the bulletin board, and that the correspondences between candidates and codenumbers also match. For print audited ballots, but not for voted ballots, the bulletin board reveals both each candidate name and the associated codenumber. Similarly, for print audited ballots, the bulletin board also unlocks the commitments published by the EA before election day associating each candidate with its codenumber. Additionally, voters and auditors may check that the information on the bulletin board is consistent with the published commitments.

Print audits are destructive: a ballot that is print audited cannot be voted. Once the commitment between candidate and codenumber is revealed, the privacy of a vote with that codenumber is compromised. By contrast, in Scantegrity III, there are two commitments for each candidate; indirection allows one of the two commitments to be revealed without compromising ballot privacy. Receipt type 1 (and its corresponding commitment) gives evidence of the cyclic shift of S1 codes. Receipt type 2 gives evidence of the S2 code for the marked position. Neither receipt alone proves the S2 code of the selected candidate.

For example, consider Figure 3. If receipt type 1 is printed, the bulletin board shows the S2 code (594) of the selected candidate (Julie) and the S1 code (B) of the the first candidate (William). It also unlocks the commitment binding the race to the S1 code (B) of the first candidate (William). If receipt type 2 is printed, the bulletin board shows the S2 code (594) of the selected candidate (Julie) and its S1 code (A). It also unlocks the commitment binding the S1 code (A) of the selected candidate to its S2 code (594).

If the station prints any receipt inconsistent with the ballot, the voter will be able to notice the discrepency in the polling place, and the receipt proves malfeasance. If the S1 code on the ballot is wrong, then with receipt type 1 only, the voter will be able to detect the discrepancy from the unlocked commitment during the on-line check. If the S2 code on the ballot is wrong, then with type 2 receipt only, the voter will be able to notice the discrepency from the unlocked commitment during the on-line check. In either case, the signed receipt proves malfeasance. Furthermore, a receipt with invalid signature also proves malfeasance, provided there is proof that the receipt came from the device. For each ballot, there is a 50% chance of being able to detect improper printing.

3.4 Policies and Procedures

With each design, some general procedures must be followed to ensure the privacy amd smooth operation of the system.

- All authorized receipt printers must be in visible locations (*e.g.*, no printer may be carried off to an undisclosed area during the election by malicious poll workers).
- The poll booth must be free of cameras, covert microphones or speakers, networking equipment or anything that can allow communication or observation between an external attacker and the voter.
- Reasonable physical security of the printers must be enforced prior to the election, heading off physical attacks against the scanning mechanism.[6]

The security of the system relies on a majority of poll workers knowing, and correctly enforcing these policies and procedures, regardless of design choice. Nevertheless, even a totally corrupt receipt printer cannot change an election outcome without detection, since any voter can compare her ballot and receipt, and any voter can verify votes on-line by checking S2 codes and by checking the unlocked commitments.

4 Requirements

The security of a receipt printer is important because, potentially, it can affect the privacy and integrity of the election. We present high-level functional and security requirements that any receipt printer for Scantegrity must implement to ensure integrity, authenticity, and confidentiality, while improving overall usability.

4.1 Functional System Requirements

Any Scantegrity printer must provide the following basic functional characteristics.

Printed Scantegrity Codes. The receipt printer will produce Scantegrity receipts, providing the user with the confirmation codes of the selected candidates and the ballot's on-line verification number.

Voter-Verification, Preferably in Polling Place. The voter must be able to verify the correctness of the receipt. It is preferable that this verification take place in the

[6] Software injection is "fair game" as it is done much more quickly than microprocessor delayering attacks

polling place, (*e.g.*, by comparing it with the marked ballot that is cast). Especially when checks are performed later on-line, it is important that the receipt provide proof to anyone of any detected malfeasance.

Self-Verifiable Receipts. For any receipt, using information printed on the receipt, anyone should be able to verify that the receipt came from an authorized receipt printer. Further, from the receipt, the voter and anyone else, shall be able to confirm that the receipt printer booted only authorized software.

Independence. The receipt printer shall not rely on the correct operation of any other component in the system.

Usability. The design should be intuitive to use and able to accommodate accessibility interfaces. A printed format may be appropriate for sighted voters, but designs should accommodate disabled voters and speakers of different languages.

Longevity. The receipt printing equipment should be able to be used in multiple elections over many years.

Failsafe. The receipt printer must not obstruct voters from being able to construct traditional handwritten receipts, including in the event of technology failure.

4.2 Security Requirements

A Scantegrity receipt printer must also uphold the following basic security properties.

Privacy. The receipt printer should not compromise voter privacy, for example, by disclosing unrevealed Scantegrity codes to unauthorized parties, disclosing the marked positions on a ballot, nor printing information on the receipt that would help correlate a confirmation code to the voter's selection.

Integrity. The receipt printer should not facilitate attacks on the integrity of the election. Further, the receipt printer should not facilitate false challenges to the election integrity. In particular, the printer shall not enable an attacker to impart credibility to a false receipt.

Event Control. The receipt printer shall not be capable of printing valid receipts before or after the authorized election period. This requirement demands strong controls on signature keys.

Information Control. Only authorized platforms shall be entrusted with any sensitive ballot data, including Scantegrity confirmation codes.

To meet these requirements, we advocate a TPM-based approach, as explained in the next section. In comparison with other approaches (*e.g.*, provisioning cryptographic material via removable smart cards), our TPM approach reduces the required trusted base and simplifies channels needed to distribute cryptographic secrets.

5 Enhancing Designs with a TPM

We propose enhancing the confidentiality of election data entrusted to the receipt printer by using a general computing platform for the receipt scanner/printer subsystem that includes a TPM. The TPM will store measurements of the device operating and application software and use these to manage keys for signing receipt data and decrypting Scantegrity II codes.

Incorporating a *Trusted Platform Module (TPM)* into the receipt printers brings three main benefits: authenticity, ballot privacy, and policy enforcement. These benefits supplement those achieved through proper procedures. For all Scantegrity III designs, the TPM can issue a digital signature of receipt contents proving legitimacy of the receipt. The TPM can enforce election day start times by protecting the signing keys with special passwords that are posted publicly only when election day begins. It also has a secure counter that can report the number of receipts issued at the end of the day, for comparison with zero tapes to detect stuffing attacks. For the marked sense translator, the Scantegrity II codes can be encrypted with a TPM key bound to the acceptable booted platform state, ensuring that incorrectly installed platforms can neither read the codes nor disclose proof of voter preferences to an attacker. The system can use encryption to maintain a secure log of all receipts scanned, so voting rights groups can verify every receipt automatically and regardless of whether individual voters verify. These benefits flow from the TPM's ability to store secrets in hardware, and bind the use of those secrets to the booted software state of the receipt printer platform.

Adding a TPM to an E2E system risks breaking the verifiability property of E2E if it were allowed to manipulate any election data (receipt code, voter preference) in a way that the voter could not verify independently. We propose using the TPM to supplement the security properties of the Scantegrity III system, adding significant design and usability improvements while being careful not to degrade the verifiability of the system. Our receipt printer design does not change the core Scantegrity system, and the voter always has the option of manually recording Scantegrity II codes for verification, independently of TPM technology.

Although the TPM protects secrets, in the mark sense translator and Scantegrity III station, it also potentially increases the risk of exposure of codenumbers beyond that caused by using invisible ink and a trusted ballot printer. Section 6.2 discusses consequences of such exposure.

5.1 Design Details

We describe the high level receipt printer enhancements with the TPM by use cases that are tied to phases of the election. The use cases involve the following hardware security features of the TPM: *Platform Configuration Registers (PCRs)* that store cryptographic hashes of receipt printer software; *Sealing*, restricting use of a secret to the identity of the TPM and the PCRs; *Monotonic Counters*, non-decreasing counters managed securely within the TPM; *Quoting*, signing the PCR values with a TPM key; *Delegation*, granting restricted administrative privilege based on knowledge of a password; and *Ownership*, the process that establishes the TPM signing and encryption key hierarchy. Further details are in Fink [18] or Carback [9]. The use cases are:

Vendor Software Delivery: The vendor delivers software to the *Independent Testing Authority (ITA)* that witnesses the final compilation. At this time, the ITA records the *golden measurements* of the software—cryptographic hashes of the operating system and receipt printer application software. The ITA securely transfers the golden measurements to the *Election Authority (EA)*.

Printer Hardware Delivery: When the receipt printers are delivered, the EA activates the TPM by *taking ownership*, causing the TPM to create a unique key hierarchy. The EA determines two passwords, an election day initiation password and an election day termination password. The EA creates the signature key, called an *Attestation Identity Key (AIK)*, sealed to the golden measurements and the usage authorization password, and publishes the public part of the AIK. The EA initiates the monotonic counter and records its known value, and creates an encryption key if using the marked sense translator. The EA creates a teardown delegation bound to the termination password.

Election Day Initialization: The EA releases the initiation password to precincts at the start of the election period. The printer boots its software, committing measurements of each booted component in the TPM's PCRs. Precinct officials gather monotonic counter values as part of the zero-tape procedures. They enter the initiation password that authorizes use of the AIK.

Voting and Receipt Printing: The voter marks her ballot and scans it as described in Section 3. The receipt printer creates a hash of the receipt data, either a signature of the scanned images in the case of the image duplicator, or a signature of the marked positions in the marked sense translator. The TPM signs the hash using the AIK and returns a quote of the platform PCRs, and the printer encodes these onto a *Two-Dimensional QR barcode (qrcode)* barcode on the receipt.

Receipt Verification: The voter takes her receipt to an independent scanning station that scans the qrcode and verifies the signature using the AIK public key. The receipt is deemed valid if the PCRs match the published golden measurements and the AIK public key is valid and known. If any problems occur, precinct officials take action.

Election Day Termination: The platform signs the final value of the monotonic counter with the AIK. The EA releases the termination password, and the poll workers enter it causing the TPM to erase its private keys so that the AIK private key never can sign anything again. The poll workers can retrieve digital archives of the receipts at poll close time, to publish for independent verification by third-parties.

As explained in Section 6.2, even if the receipt printer were malicious with corrupted TPM, the adversary could not change the election outcome without detection. Moreover, the Scantegrity II voting system with any of our receipt printers remains software independent: no undetected fault in software can change the election outcome without detection [36, 35].

5.2 Assumptions

Our TPM design enhancements use the hardware protection features of the TPM to protect election secrets, enforce election policy, and protect voter privacy by keeping secrets away from malicious software. However, the TPM has its limits, as does any high value system that uses sophisticated electronic and physical components.

In particular, if the attacker controls the TPM or its lifecycle, or if the attacker controls the software lifecycle and is able to insert back-door software activation codes into the reviewed and certified election software, the TPM's protections become useless. Control over the TPM or election software lifecycle by an adversary would allow these conditions to occur (countered by certain mitigations):

- The system could authenticate fake receipts. The remedy is detecting this situation when the voter notices incorrect Scantegrity E2E codes on the public bulletin board and raises a challenge. Although the receipt seems authentic, election authorities may recover the original ballot and confirm the TPM or software malfunction. This problem would be exacerbated if the attacker also controlled the physical ballots, encoding them with special disappearing inks that would reveal one code during voting and fade out to become a different code, chemically morphed to match the fake receipt.

- The system could refuse to authenticate legitimate receipts. The remedy is detecting this condition in the precinct; the Scantegrity E2E codes would be verified at the public bulletin board.

Acknowledging these threats, we claim that our TPM design can authenticate valid receipts and detect malicious software that booted only when certain assumptions hold:

1. The attacker does not control the TPM or its lifecycle.
2. The attacker does not control the lifecycle of the certified election software.
3. The attacker cannot influence the EA to disclose passwords prematurely, including the election initiation and TPM ownership passwords.
4. The attacker does not control the root of trust for measurement, that portion of the *Basic Input/Output System (BIOS)* or CPU responsible for initiating the sequence of measurements that occurs during platform boot.[7]

We trust the measured boot chain to include the entire software image of the receipt printer, plus the critical OS software functions and supporting libraries. This measured boot chain typically begins with the BIOS, although Intel and AMD support a capability called Late Launch that allows late booting into a secure environment. The boot sequence is responsible for storing all the software measurements in the TPM. If the boot sequence is improperly designed or inadequately tested prior to deployment, it may skip or incorrectly record one or more measurements causing the TPM not to reflect the state of all software in control of the receipt printer. This can enable an adversary to take control of the platform even though the recorded state appears to be correct. A similar problem can happen if the attacker is in control of the System Management Mode as described below.

Subverting the TPM and trusted boot requires a sophisticated attack, yet is within the capabilities of major nation states. Wojtczuk and Rutkowska [43] attacked Intel's TXT trusted boot process by injecting code into the privileged System Management Mode (SMM) feature on Intel processors that corrupts the kernel image immediately after the TPM records its measurement, invalidating the trusted measurement chain. Intel has proposed a fix for this attack, and finding new exploits in SMM is difficult by the authors' own admission. Tarnovsky [41] demonstrated how to delayer and probe the unencrypted data paths of the Infineon TPM, but this attack requires persistent physical access and time-consuming instrumentation. Sophisticated attackers may also find it possible to learn codes by defeating the invisible ink, compromising the trusted ballot printer, or compromising the trusted workstation that generates the master election secret.

5.3 Risks

As explained in detail in Section 6.2, our receipt printers introduce additional potential risks to privacy but do not enable an attacker to modify an election outcome without detection, even if the receipt printer and its TPM are completely compromised. For example, in the mark sense translator and Scantegirty III station, compromise of the TPM leaks the device's signature key and exposes all Scantegrity II verification numbers read by that device. This exposure enables the attacker to violate ballot privacy and make false claims of incorrectly posted codenumbers. The situation is similar to, and slightly worse than, compromise of codenumbers in Scantegrity II through failure of the central printer, invisible ink, or custody of unmarked ballots shipped from the printer to the precincts. Similarly, in traditional Scantegrity II, a corrupt scanner can also expose the codenumbers it reads.

6 Discussion

In this section we discuss several issues arising from our receipt printer designs. In particular, we discuss security benefits from trustworthy receipt printers, threats (and their mitigations) from malicious receipt printers, and other issues.

6.1 Security Benefits

In addition to improving usability through automation, our receipt printers offer significant security benefits.

Trustworthy receipt printers mitigate the threat of someone (*e.g.*, a malicious poll worker with access to marked ballots) adding marks to a cast ballot. For example, an added mark to an under-voted race could help a candidate, and an added over-vote to a race could invalidate a vote for another candidate. A digitally-signed receipt offers proof of such malfeasance. By contrast, in Scantegrity II, a voter cannot prove that she under-voted because she cannot prove that she does *not* know any of codes. Similarly, she cannot prove that she did *not* over-vote.

Receipt printers enable the Election Authority (EA) to produce an electronic and/or paper copy of all receipts printed. These receipts could be posted publicly and given to various auditors. In doing so, auditors could verify all votes.

Because receipt printers make it easier for voters to generate receipts, it seems likely that more voters will verify their votes on-line.

[7] TPM and CPU manufacturers have studied this problem in depth; in particular, Intel's AC_INIT CPU extension securely initiates the measurement process and forms a core part of their Trusted eXecution Technology architecture. See Grawrock [22] for details.

The Scantegrity III system achieves the security benefit of full voter verifiability. By contrast, Scantegrity II voters depend on print audits to establish statistical assurance that ballots are correctly printed. While the voter can void and audit her Scantegrity II ballot, she cannot audit her cast ballot.

6.2 Malicious Receipt Printers

Receipt printers also present potential security vulnerabilities, especially—but not limited—to voter privacy. We now consider what might happen if an attacker gained control of our receipt printers. After explaining our threat model, we analyze a variety of potential attacks.

6.2.1 Threat Model

We consider only attacks that use a rogue receipt printer during the election. For a more general security analysis of Scantegrity and voting, see [14, 24, 25]. For an analysis of TPM protocols for voting, see [19]. Our adversary might include an insider, a foreign government, a minority of corrupt poll workers, one or more of the contestants, or a coerced or paid voter.

We consider four categories of attack:

1. *Manipulation Attacks*, where an adversary attempts to manipulate the election result.

2. *Identification Attacks*, where an adversary attempts to identify voter choices and violate election privacy.

3. *Disruption Attacks*, where an attacker wishes to prevent or delay certification of the election undetectably.

4. *Discreditation Attacks*, where an attacker tries to imbue significant doubt in the public's perception of the election outcome.

We do not consider denial of service attacks explicitly, because they apply to all voting systems and are difficult to prevent but typically are easy to detect. A receipt printer's security design is successful if it does not increase the ability of an adversary to carry out successful attacks undetectably, in comparison with not using the receipt printer.

6.2.2 Manipulation Attacks

Importantly, each of our designs enables the voter to compare her receipt against her marked ballot. Furthermore, she may additionally create her own handwritten receipt. Therefore, a corrupt receipt printer cannot modify the election outcome without detection.

A malicious receipt printer could perform a *chain printing* attack, where it tires to trick the voter into checking the wrong confirmation code on-line. The receipt printer saves and reprints from a previous ballot valid scanned images (or verification codes) and corresponding ballot on-line verification number. The conspiring PCOS scanner flips votes of the unverified ballots, altering the election outcome. After leaving the polling place, the voter will be unable to detect malfeasance when verifying her vote on-line. A voter can detect chain printing by comparing her marked ballot with the printed receipt in the polling place. Auditors can also check if the receipt printer prints the same on-line verification number more than once.

Scantegrity is potentially vulnerable to a *ballot-stuffing* attack, since voters cannot directly check if the official data include extra ballots. This attack is detected by poll workers keeping a careful count of how many people voted and checking that the data do not include extra marked ballots. Because it is not connected to the PCOS scanner, the image duplicator provides no defense against this attack. The mark sense translator offers a TPM-protected monotonic counter printed on the receipt as an independent count of the number of ballots cast. But a corrupt mark sense translator could either not count the stuffed ballots inserted into the scanner (hoping that poll workers will not count voters accurately), or it could count them but not print a corresponding receipt (hoping that poll workers will have more confidence in the TPM's monotonic counter over their own counts, and hoping that voters will not notice the discountinuity in receipt sequence numbers).

6.2.3 Identification Attacks

Protecting privacy is very difficult. For example, a corrupt receipt printer could photograph the voter and transmit a cryptographically signed image of the marked ballot via a clandestine channel. But this threat also exists with conventional PCOS scanners. Our main privacy protection stems from our use of a TPM to store keys and verification codes and to verify that the correct software was booted.

Because the receipt printers scan marked ballots, they see sensitive information (as do PCOS scanners), including marked choices and their codes. Furthermore, the mark sense translator knows all codes on each ballot scanned. This sensitive information could be leaked through clandestine channels (*e.g.*, through signatures or subtle steganographic modifications to the printed fonts).

The attacker must still associate the voters with their marked ballots. One way to do so is via the "Italian" attack, where the coercer demands that the voter mark certain unimportant races in distinctive ways. Another way is with the aid of a corrupt image duplicator. Since the image duplicator prints partially marked ovals, the coercer could demand that the voter partially mark ovals in certain races in distinctive ways.

A malicious image duplicator could also scan the marked ballot at a slight angle, revealing the correspondence between codes and their candidates.

A malicous image duplicator might try to leak ballot choices by selecting the permutation of ovals on the receipt based on the on-line verification number. However, if ovals are not printed in order of decreasing pixel density, the receipt is proof of malefeasance.

If voters tend to fill in ovals more completely depending on the oval position on the ballot, then the image duplicator might leak some information about candidate choices.[8]

A stronger architectal defense, but requiring more radical changes to the voter experience, would be to separate the acts of voter intent capture and casting, as in the Benaloh model [6], and to permit multiple intent captures. Then, any scanner/encryptor seeing a plaintext ballot would not know if that ballot were cast, and only encrypted ballots would be cast.

6.2.4 Disruption and Discreditation Attacks

These attacks are also very hard to prevent. For example, a machine that misbehaves in various ways (*e.g.*, jams paper, prints unreadable receipts, stops working) can cause delays and lower public confidence in the system. But this threat also exists with conventional scanners. Regardless, the EA has the option to examine the marked paper ballots.

There are many ways in which a corrupt receipt printer (or PCOS scanner) could potentially misbehave in detectable ways that cause disruption and thereby errode voter confidence in election results. For example, the corrupt receipt printer could print incorrect codes or online verification numbers, generate invalid digital signatures, print additional valid codenumbers of *unmarked* choices, or carry out detectable manipulation attacks.

Conversely, a malicious voter could forge a receipt with *invalid* signature and falsely claim that the receipt printer generated the bogus receipt. Even if the voter were allowed to check digital signatures in the polling place (*e.g.*, with her iPhone), it would be nearly impossible to prevent a voter from smuggling a bogus receipt into the polling place. To mitigate this threat, the EA might print receipts onto security paper, but doing so adds cost and complexity and does not provide a perfect defense, including against insiders with access to the security paper.

If the receipt printer leaks a valid codenumber of an unvoted choice, the conspiring malicious voter could convincingly falsely claim that the on-line bulletin board posted the wrong verification code.

Despite these threats, our receipt printers do not significantly worsen the current threat from malicious scanners, and the TPM helps ensure that the correct software is booted.

6.3 Eliminating Invisible Ink

Fink [18] and Carback [9] speculated that the mark sense translator could be modified to eliminate the need for invisible ink in Scantegrity ballots by entrusting the Scantegrity II codes to the cryptographic protections of the TPM, dividing the election codes into groups and encrypting them with unique secrets shared between specified TPMs and the election authority. This capability would enable the printer to“late bind” the codes to the marked ballot positions, recording the codes in the clear for the first time only on the printed receipt. Such an approach eliminates the complexity of invisible ink ballots, hides the codes until they are needed (thus mitigating several privacy attacks based on access to unvoted ballots), and improves accessibility, *e.g.*, by allowing blind voters to hear codes on accessibility devices.

Despite these advantages, some argue that this strategy places unwarranted trust in the TPM, making it the gatekeeper of critical Scantegrity II codes. As a safeguard, the ballots can also include Scantegrity I codes. Nevertheless, failures in the operation or integrity of the TPM would compromise voter privacy and reduce the ballot verification properties of Scantegrity II (using late binding) to that of Scantegrity I. On the other hand, traditional Scantegrity II trusts the central ballot printer operation not to reveal codes, and it trusts the integrity of the physical ink process and chain of custody between the central printer and the precincts. While we agree that invisible ink is the best present alternative, we feel late binding is worth further investigation because it is easy to do using a TPM, potentially offers more granular safeguards of the election secrets, and affords an option to those election authorities or specific circumstances where invisible ink is impractical.

7 Conclusion

We have presented and analyzed three designs for a trustworthy receipt printer for Scantegrity II, representing different engineering tradeoffs among simplicity, usability, and security.

The image duplicator is attractive for its simplicity and because it can augment exisitng PCOS scanners as a separate optional station without any modifications to the

[8] Observation by an annonymous referee.

PCOS scanners. It can be used in addition to either of the other designs. As a separate, optional station that is not provided with codenumbers, this design offers the least additional security risk over that already present in Scantegrity II. The image duplicator, however, offers no guarantee that the ballot presented to the receipt printer is the same ballot cast, and it requires an extra station in the voting process. Nevertheless, it is our favorite choice.

The mark sense translator guarantees that the receipt is based on the cast ballot. By knowing the confirmation codes (*e.g.*, as encrypted on the ballot), this design can print the codes more clearly, offer more accessibility options, and provide a meaningful check on the PCOS scanner. The translator, however, requires a physical mechanism to lock the ballot under glass after scanning and before casting, and it creates a greater potential security vulnerability by having access to all codenumbers on the ballots it scans.

As an embellished mark sense translator, the Scantegrity III casting station can backlight ballots to point out important feartures, including over- and under-votes. It eliminates the need for a separate print audit. But it is the most complex of the three designs, uses a more complicated ballot with both Scantegrity I and II codes, and (for voters who wish to verify their ballots) requires voters to carry out a more involved checking procedure at the station. The Punchscan-like indirection of the Scantegrity I codes will likely confuse some voters. The intriguing backlighting user interface is of separate interest.

The mark sense translator and Scantegrity III station offer some security advantages (*e.g.*, empowering voters to detect ballot modifications after casting), but they also present additional security risks (*e.g.*, exposure of codenumbers through the TPM). Those who find such risks unacceptable will likely prefer the image duplicator.

Each design depends on a sufficient number of voters comparing the printed receipt with the marked ballot (and/or making their own handwritten receipt). This situation is far better than that of VVPAT: Our designs preseve E2E outcome integrity. And while the usability for the voter remains to be tested, election officials do not have to hand-count VVPAT printouts.

Open problems include implementation, usability testing, and adding accessibility interfaces. It remains to be determined how easily voters can read the codenumbers printed by the image duplicator, and how voters will respond to the Scantegirty III backlighting feature. It also remains to be determined how well voters will deal with the conceptual and physical complications of the Scantegrity III casting station, despite other simplifying and attractive aspects of its user interface.

All designs facilitate increased verification of confirmation codes by making it easier for voters to bring home receipts and by enabling the election authority to release copies of all receipts generated. Because the receipt printers scan marked ballots, they introduce potential security vulnerabilities, especially to voter privacy and to supporting false claims of irregularities. Using the TPM as a trusted base helps voters verify that the platforms booted the correct software, that the receipts are genuine, and that voter privacy is maintained. In each design, the voter verifies the paper receipt in the polling place by comparing it against the marked ballot. Furthermore, voters are welcome to make their own additional handwritten receipts. Consequently, these receipt printers cannot change election outcomes without detection.

8 Acknowledgments

We appreciate helpful comments from Josiah Dykstra, Christopher Nguyen, Ronald Rivest, Vanessa Teague, and the reviewers. Sherman was supported in part by the Department of Defense under IASP grants H98230-09-1-0404 and H98230-10-1-0359.

References

[1] 42ND CONGRESS OF THE UNITED STATES OF AMERICA. The Help America Vote Act of 2002 (HAVA). United States Public Law 107-252, 2002.

[2] ADIDA, B. *Advances in Cryptographic Voting Systems*. PhD thesis, MIT, August 2006.

[3] ADIDA, B. Helios: Web-based open-audit voting. In *Proceedings of the 17th Usenix Security Symposium (USENIX Security 2008)* (July 2008), USENIX Association, pp. 335–348.

[4] ADIDA, B., AND NEFF, C. Efficient Receipt-Free Ballot Casting Resistant to Covert Channels. In *EVT/WOTE'09: Proceedings of the USENIX Electronic Voting Technology Workshop/Workshop on Trustworthy Elections* (Berkeley, CA, USA, 2009), USENIX Association.

[5] ADLER, J., DAI, W., GREEN, R., AND NEFF, C. Computational details of the votehere homomorphic election system. In *Proc. Ann. Intl Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT)* (2000).

[6] BENALOH, J. Simple verifiable elections. In *EVT'06: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop* (Berkeley, CA, USA, 2006), USENIX Association.

[7] CARBACK, R., CHAUM, D., CLARK, J., CONWAY, J., ESSEX, A., HERRNSON, P. S., MAYBERRY, T., POPOVENIUC, S., RIVEST, R. L., SHEN, E., SHERMAN, A. T., AND VORA, P. L. Scantegrity II Municipal Election at Takoma Park: The First E2E Binding Governmental Election with Ballot Privacy. In *19th USENIX Security Symposium* (Washington, DC, USA, August 2010), USENIX Association.

[8] CARBACK, R., CHAUM, D., CLARK, J., CONWAY, J., ESSEX, A., HERRNSON, P. S., MAYBERRY, T., POPOVENIUC, S., RIVEST, R. L., SHEN, E., SHERMAN, A. T., VORA, P. L., AND SINHA, B. Exploring Reactions to Scantegrity: Analysis of Survey Data from Takoma Park Voters and Election Judges. Pending Publication, 2010.

[9] CARBACK, R. T. I. *Engineering Practical End-to-End Verifiable Voting Systems*. PhD thesis, Dept. of CSEE, University of Maryland, Baltimore County, Baltimore, MD, 2010.

[10] Challener, D., Yoder, K., Catherman, R., Safford, D., and Van Doorn, L. *A Practical Guide to Trusted Computing*. IBM press, Upper Saddle River, NJ, 2007. ISBN 978-0132398428.

[11] Chaum, D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM 24*, 2 (1981), 84–90.

[12] Chaum, D. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security and Privacy 2*, 1 (2004), 38–47.

[13] Chaum, D., Carback, R., Clark, J., Essex, A., Popoveniuc, S., Rivest, R. L., Ryan, P. Y. A., Shen, E., and Sherman, A. T. Scantegrity II: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In *EVT'08: Proceedings of the Conference on Electronic Voting Technology* (Berkeley, CA, USA, 2008), USENIX Association, pp. 1–13.

[14] Chaum, D., Carback, R., Clark, J., Essex, A., Popoveniuc, S., Rivest, R. L., Ryan, P. Y. A., Shen, E., Sherman, A. T., and Vora, P. L. Scantegrity II End-to-End Verifiability by Voters of Optical Scan Elections Through Confirmation Codes. *IEEE Transactions on Information Forensics and Security: Special Issue on Electronic Voting* (2009).

[15] Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A., and Vora, P. Scantegrity: End-to-end voter-verifiable optical-scan voting. *IEEE Security and Privacy 6*, 3 (2008), 40–46.

[16] Chaum, D., Ryan, P. Y. A., and Schneider, S. A. A practical, voter-verifiable, election scheme. Technical Report Series CS-TR-880, University of Newcastle Upon Tyne, School of Computer Science, December 2004.

[17] Fink, R., and Sherman, A. Combining end-to-end voting with trustworthy computing for greater privacy, trust, accessibility, and usability (summary). In *Proceedings of the National Institutes of Technology (NIST) workshop on end-to-end voting systems* (October 13-14 2009).

[18] Fink, R. A. *Applying Trustworthy Computing to End-To-End Electronic Voting*. PhD thesis, Dept. of CSEE, University of Maryland, Baltimore County, Baltimore, MD, 2010.

[19] Fink, R. A., Sherman, A. T., and Carback, R. TPM meets DRE: Reducing the trust base for electronic voting using trusted platform modules. *IEEE Transactions on Security and Forensics 4*, 4 (2009), 628–637.

[20] Fisher, K., Carback, R., and Sherman, A. Punchscan: Introduction and system definition of a high-integrity election system. In *Preproceedings of the 2006 IAVoSS Workshop on Trustworthy Elections (WOTE 2006)* (Robinson College, Cambridge, United Kingdom, June 2006). Available at www.punchscan.org/papers/fisher_punchscan_wote2006.pdf.

[21] Goggin, S. N., Byrne, M. D., Gilbert, J. E., Rogers, G., and McClendon, J. Comparing the auditability of optical scan, voter verified paper audit trail (VVPAT) and video (VVVAT) ballot systems. In *EVT'08: Proceedings of the conference on Electronic voting technology* (Berkeley, CA, USA, 2008), USENIX Association, pp. 1–7.

[22] Grawrock, D. Dynamics of a Trusted Platform: A building block approach.

[23] IBM Corporation. The Trusted Computing Software Stack (TrouSerS) software library. Available at http://sourceforge.net/projects/trousers/, 2008. Last accessed Feb 3, 2011.

[24] Kelsey, J., Regenscheid, A., Moran, T., and Chaum, D. Attacking Paper-Based E2E Voting Systems. *Towards Trustworthy Elections* (2010), 370–387.

[25] Küsters, R., Truderung, T., and Vogt, A. Proving Coercion-Resistance of Scantegrity II. In *Proceedings of the 12th International Conference on Information and Communications Security (ICICS 2010)*, vol. 6476 of *Lecture Notes in Computer Science*. Springer, 2010, pp. 281–295.

[26] Kutyowski, M., and Zagrski, F. Verifiable internet voting solving secure platform problem. In *Advances in Information and Computer Security*, A. Miyaji, H. Kikuchi, and K. Rannenberg, Eds., vol. 4752 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, pp. 199–213. 10.1007/978-3-540-75651-4%5F14.

[27] Lohr, C. Semantic Light: Building Blocks. MS Thesis, Dept. of CSEE, University of Maryland, Baltimore County. Available at http://cnlohr.net/projects/semanticlight/, May 2011.

[28] Neff, C. A. Practical high certainty intent verification for encrypted votes, 2004.

[29] Norris, D. F. P., Sears, A. C., Nicholas, C. C., Roland, A. V. E., Gangopadhyay, A., Holden, S. H., Karabatis, G., Koru, A. G., Law, C. M., Pinkston, J., Sherman, A. T., and Zhang, D. A study of vote verification technologies. part I: Technical Study. Prepared for the Maryland State Board of Elections, National Center for the Study of Elections of the Maryland Institute for Policy Analysis and Research, University of Maryland, Baltimore County, February 2006.

[30] Norris, D. F. P., Sears, A. C., Nicholas, C. C., Roland, A. V. E., Gangopadhyay, A., Holden, S. H., Karabatis, G., Koru, A. G., Law, C. M., Pinkston, J., Sherman, A. T., and Zhang, D. A study of vote verification technologies. part I: Technical Study Appendices. Prepared for the Maryland State Board of Elections, National Center for the Study of Elections of the Maryland Institute for Policy Analysis and Research, University of Maryland, Baltimore County, February 2006.

[31] Pearson, S., and Balacheff, B. *Trusted Computing Platforms: TCPA Technology in Context*. Prentice Hall PTR, 2003.

[32] Popoveniuc, S., and Hosp, B. An introduction to punchscan. In *Proceedings of the 2006 IAVoSS Workshop on Trustworthy Elections* (2006).

[33] Popoveniuc, S., Kelsey, J., and Regenscheid, A. Performance requirements for end-to-end verifiable elections. In *EVT/WOTE'10: Proceedings of the Electronic Voting Technology Workshop/Workshop on Trustworthy Elections* (Berkeley, CA, USA, 2010), USENIX Association/IAVoSS/ACCURATE, p. 16.

[34] Popveniuc, S., and Regenscheid, A. Sigma Ballots. In *EVOTE2010: The 4th International Conference on Electronic Voting* (Bregenz, Austria, July 2010), E-Voting.CC.

[35] Rivest, R., and Wack, J. On the notion of "software independence" in voting systems. DRAFT Version Retrieved on September 25, 2007.

[36] Rivest, R. L. On the notion of 'software independence' in voting systems. *Philosophical Transations of the Royal Society 366*, 10.1098/rsta.2008.0149 (October 2010), 3759–3767.

[37] Sandler, D. R., Derr, K., and Wallach, D. S. VoteBox: a tamper-evident, verifiable electronic voting system. In *Proceedings of the 17th Usenix Security Symposium* (2008).

[38] Sherman, A., Gangopadhyay, A., Holden, S., Karabatis, G., Koru, A., Law, C., Norris, D., Pinkston, J., Sears, A., and Zhang, D. An examination of vote verification technologies: Findings and experiences from the Maryland Study. In *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop* (2006), USENIX Association, p. 10.

[39] SHERMAN, A. T., CARBACK, R., CHAUM, D., CLARK, J., ESSEX, A., HERRNSON, P. S., MAYBERRY, T., POPOVENIUC, S., RIVEST, R. L., SHEN, E., SINHA, B., AND VORA, P. L. Scantegrity Mock Election at Takoma Park (summary). In *Workshop on End-to-End Voting Systems* (Washington, DC, USA, October 2009), National Institute of Standards and Technology.

[40] SHERMAN, A. T., CARBACK, R., CHAUM, D., CLARK, J., ESSEX, A., HERRNSON, P. S., MAYBERRY, T., POPOVENIUC, S., RIVEST, R. L., SHEN, E., SINHA, B., AND VORA, P. L. Scantegrity Mock Election at Takoma Park. In *EVOTE2010: The 4th International Conference on Electronic Voting* (Bregenz, Austria, July 2010), E-Voting.CC.

[41] TARNOVSKY, C. Deconstructing a 'Secure' processor. In *Black Hat DC* (2010).

[42] TRUSTED COMPUTING GROUP. TCG TPM specification version 1.2, revision 103. Available at https://www.trustedcomputinggroup.org/specs/TPM, 2008. Last accessed on Mar 15, 2008.

[43] WOJTCZUK, R., AND RUTKOWSKA, J. Attacking Intel® Trusted eXecution Technology. *Black Hat DC* (2009).

A Additional Design Details

In this section we provide additional design and use details for the image duplicator and mark sense translator.

A.1 Image Duplicator

Important design details of the image duplicator include how the markable positions–the ovals–are identified and scanned, and how the oval images are presented.

Scanning Markable Positions. The image duplicator identifies markable positions within individual contests using (x, y) offsets from preprinted alignment marks detected on the ballot. The image duplicator follows Scantegrity's practice of additionally using dark circles on the ballot to identify the qrcode and the markable positions to the PCOS scanner [15].

The image duplicator scans an image of the entire area of each markable position. We suggest a scan resolution of 150 dots per inch and 8-bit grayscale (256 levels of gray), which is sufficient to resolve revealed Scantegrity codes. The image duplicator does not attempt to determine the filled state of any oval (*i.e.*, whether or not it is marked), but merely captures the image as presented on the ballot.

Printing Scanned Ovals. The image duplicator groups the images of the scanned markable positions by contest, and sorts the images within each contest by average pixel value. The average pixel value is computed over an 8-bit grayscale representation of the image. Thus, images of fully marked ovals appear first, followed by partially marked ovals, followed by blank ovals. For each contest, the image duplicator prints onto the receipt a contest indicator, *e.g.*, "contest 1", and the scanned oval images for that contest. To make the receipt easier to read, the duplicator can enlarge the printed images.

Checking the Receipt. To verify that the receipt is correctly and intelligibly recorded, the voter compares it with the marked ballot. She verifies that the on-line verification number and the codenumbers on the receipt match the corresponding ones on the ballot. It is helpful for the voter to check the legibility of the receipt at this step before leaving the polling place.

A.2 Mark Sense Translator

Critical design details of the mark sense translator include contents of the receipt, protection of the Scantegrity verification codes, and required modifications to the PCOS scanner

Receipt Contents. As with the image duplicator, the mark sense translator groups the revealed codes by contest. Within each contest, it verifiably randomly orders the codes of all marked ovals. Unlike the image duplicator, the mark sense translator does not scan an image of any oval, and therefore it does not report codes of partially marked ovals that are not sensed by the PCOS scanner as marked positions.

Connection to PCOS Scanner. The PCOS scanner is connected to the mark sense translator using a data cable. The data sent to the mark sense translator include: (a) on-line verification number; (b) contest designations; (c) marked positions by contest (*e.g.*, "contest 1, position 1 of 3 is marked"); and (d) optional indication of over- or under-voting per contest.

The PCOS scanner enforces a well-defined message interface format to protect it from a corrupt mark sense translator that may send ill-formed messages to the PCOS. A one-way data cable may mitigate this threat. Regardless, as explained in Section 6.2, corrupt scanners and receipt printers cannot change the election outcome without detection.

Scantegrity Codes Retrieval. As explained in Section 5, the receipt printer uses keys stored on its TPM to decrypt Scantegrity codes corresponding to marked positions. These encrypted codes can be communicated to the receipt printer via the ballot's qrcode or through a special channel at election day setup.

Each qrcode printed on Scantegrity ballots can encode up to 2,953 binary bytes, enough for about 1,400 individual 3-digit codes. Transporting the codes with the ballot reduces pre-election work, and requires only a single chain of custody for both the physical ballot and its digital representation.

Required Modifications to the PCOS Scanner. The PCOS must be modified to supply the on-line verification numbers and sensed marked positions to the receipt printer. These data are part of the content already retained by the PCOS scanner.