Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes

David Chaum Richard Carback* Jeremy Clark† Aleksander Essex‡
Stefan Popoveniuc§ Ronald L. Rivest¶ Peter Y. A. Ryan‖ Emily Shen¶
Alan T. Sherman*

Abstract

We introduce Scantegrity II, a practical enhancement for optical scan voting systems that achieves increased election integrity through the novel use of confirmation codes printed on ballots in invisible ink. Voters mark ballots just as in conventional optical scan but using a special pen that develops the invisible ink. Verifiability of election integrity is end-to-end, allowing voters to check that their votes are correctly included (without revealing their votes) and allowing anyone to check that the tally is computed correctly from the included votes. Unlike in the original Scantegrity, dispute resolution neither relies on paper chits nor requires election officials to recover particular ballot forms. Scantegrity II works with either precinct-based or central scan systems. The basic system has been implemented in open-source Java with off-the-shelf printing equipment and has been tested in a small election.

An enhancement to Scantegrity II keeps ballot identification and other unique information that is revealed to the voter in the booth from being learned by persons other than the voter. This modification achieves privacy that is essentially equivalent to that of ordinary paper ballot systems, allowing manual counting and recounting of ballots.

1 Introduction

> "Trust, but verify."
>
> *—Russian proverb, popularized by Ronald Reagan during his presidency*

In order to provide increased confidence in the integrity of election outcomes, modern voting technology and legislation is moving from a model of trust to one that provides verifiability[1] of some steps in the voting process. For example, with optical scan systems and voter-verified paper audit trails (VVPAT), voters can verify that their votes are correctly recorded on paper, and post-election manual recounts can verify that the paper trail matches the electronically produced tally. *End-to-end (E2E)* voting systems[2] provide additional, stronger verifiability. E2E systems allow voters to verify that their ballots are processed correctly, giving voters assurance that their votes are cast, collected, and counted as intended. In essence, end-to-end voting systems provide a verifiable chain of custody on the ballots, from the start, when the votes are cast, to the end, when they are counted. They also enable the correctness of the final tally to be verified by anyone.

We propose Scantegrity II: **I**nvisible **I**nk, an end-to-end security enhancement for existing optical scan voting systems that enables each voter to verify that her vote is processed correctly, without introducing any new polling place equipment. Extra information is printed on optical scan ballots during production, but the underlying method by which voters mark the ballots is not changed and remains in accordance with legislative proposals requiring "unencrypted" paper audit records. The E2E verifiability of Scantegrity II is opt-in; the voters who choose to ignore the additional features will have a voting experience similar to that with current optical scan. Additionally, Scantegrity II does not interfere with

*University of Maryland, Baltimore County, USA
†University of Waterloo, Canada
‡University of Ottawa, Canada
§The George Washington University, USA
¶Massachusetts Institute of Technology, USA
‖University of Newcastle upon Tyne, United Kingdom

[1] It is often useful to distinguish "verifiability" from "verification." Some systems may legally compel a voter to verify certain aspects of her vote prior to casting it. "Verifiability" herein refers to the ability to verify and not the legal status of being verified.

[2] The term "voting system" has traditionally referred to the algorithm for determining an election outcome; however, recently in the computer science literature, it has been used to refer to various subsystems in the complex system of voting. We follow the latter consensus herein.

the security provisions of the underlying optical scan system: physical observation, validated tallying software, and recounts.

Contributions Scantegrity II inherits several contributions from the original Scantegrity.

- End-to-end verifiability of election integrity: Scantegrity II allows voters to verify that their choices are included unmodified in the final tally and allows anyone to verify that the collection of included choices is counted properly.
- Compatibility with optical scan equipment: Scantegrity II is an add-on and does not require any optical scan polling place equipment to be replaced. It interfaces cleanly with the underlying optical scan system, requiring only a modified ballot and access to the results from the scanners.
- Familiar ballot marking procedure: the voter marks the bubble beside the candidate she wishes to vote for, just as she would on a conventional optical scan ballot. Opting into verifying the strong integrity that Scantegrity II provides is up to the individual voter.

The basic Scantegrity II scheme, presented in Sections 3 through 6, provides the following new contributions.

- Immunity to coercion and "randomization" attacks: when a voter looks at an unmarked Scantegrity II ballot, the information she will retain as a receipt is hidden. Therefore, an attacker cannot force a voter to mark her ballot in a way that creates a prespecified receipt, unlike in some other E2E systems including the original Scantegrity.
- An improved mechanism for handling disputes: since confirmation codes are printed in invisible ink, the election officials can distinguish between serious claims of discrepancies and spurious ones without resorting to recovery of particular physical ballots, as required in the original Scantegrity. Election officials set statistical triggers that detect fraud while minimizing false positives.

The enhanced version of Scantegrity II, presented in Section 7, offers further contributions.

- The use of special ink to make ballots appear humanly indistinguishable from each other after they are cast preserves voter privacy during manual recounts or any other inspection of the paper ballots—whether authorized or not.
- An informational mechanism to provide the voter with undeniable proof of any error in the posting on the election website without relying on possession of a physical document, as does the basic scheme.

Organization In the following section, we discuss work related to our proposal, including an overview of the original version of Scantegrity. Section 3 provides a high-level overview of the voter experience. In Section 4, we provide a detailed chronology of each step in a Scantegrity II election. We discuss the security of the system in Section 5 and some implementation details about using invisible ink in Section 6. In Section 7 we present enhancements to the basic Scantegrity II system of Sections 3 through 6.

2 Related Work

An end-to-end (E2E) approach to voting strives to provide verifiability of the integrity of key steps in the voting process. In particular, it allows verification that all cast ballots are included unmodified in the collection of ballots that are counted—a property assured by neither voter-verified paper audit trails (VVPAT) nor manual recounts. More formally, what have been variously called *E2E*, *coded vote*, *cryptographic*, or *open-audit voting systems*, are systems which preserve ballot secrecy while providing:

1. *Voter Verifiability:* some time after casting her ballot, each voter can confirm that her vote was "collected as cast" by checking privacy-preserving receipt information against a public record of receipts posted by the election officials.
2. *Universal Verifiability:* anyone can verify that votes were "counted as collected", *i.e.*, the posted tally is correct with respect to the public record of posted receipts.

Cryptographic techniques were first applied to voting by Chaum [5], and Scantegrity II is a descendant of this paradigm [7, 10, 6]. An earlier paper-based system by Chaum [7] provides voter verifiability through a two-sheet ballot printed by a special printer that uses a transparent sheet and visual cryptography to show voter choices in a human-readable format. The voter destroys one sheet and keeps the other as a privacy-preserving receipt, a copy of which is publicly posted. It provides universal verifiability using a process similar to a mixnet [5] to decode the receipts and recover the plaintext ballots in an unlinkable manner.

The Punchscan [16, 22] and Prêt à Voter [9] systems use simpler printing. In both systems, positional data

from a printed ballot is publicly posted for voter verifiability. Each voter uses information printed on the unique ballot that indicates which positions correspond to which candidates in order to mark the ballot and subsequently destroys part of that information in order to hide her choices. To provide universal verifiability, the original Prêt à Voter uses a mixnet [5] and Punchscan uses a specialized anonymity network called the Punchboard, which combines tables of binding commitments [20, 4] with randomized partial checking (RPC) [17]. It is now recognized, however, that such particular universal verifiability "system back-ends" can be mixed and matched with "voter-facing front-ends." [19, 24]

Both Punchscan and Prêt à Voter have numerous desirable security properties [25, 14]. One potential issue, however, with any such system that publishes per-ballot contest results, including these two as well as Scantegrity, is known as the "Italian attack" or "pattern voting." This has been solved for inter-contest patterns using so-called "contest partitioning" [23]. Another issue with these previous systems is so-called "randomization attacks" [15, 18], in which a coercer can verify whether a voter has in effect randomized or biased the vote [18]. As mentioned earlier, these randomization attacks are solved by Scantegrity II.

The usability of the ballot marking process in any voting system can affect the accuracy of capturing voter intention. The indirect association of candidate order and mark order of the Punchscan ballot may conflict with the voter's mental model of how a ballot should be marked. On the other hand, voting systems requiring additional thought related to marking mechanics can increase the accuracy with which some voters mark. The experience of some of the authors using Punchscan in both mock and binding elections found no significant problems with the indirection [13].

In any case, Scantegrity [8] sidesteps these potential usability shortcomings by using a standard, non-randomized candidate ordering with adjacent bubbles, such as with conventional optical scan ballots. Each ballot independently associates a code with a particular candidate. The receipt therefore is simply a list of the codes of the selected candidates. This change interfaces directly with the pre-existing Punchscan anonymizing network [21], but election officials must recover individual paper ballots to settle disputes, and this leads to a cumbersome dispute resolution process in order to preserve voter privacy. The improvements presented in this paper eliminate the need for any such physical dispute resolution process.

The use of invisible ink is related to the concept of ballot casting assurance [1], where correctness of the receipt is verified by giving a voter a choice to audit it after it is created but before it is seen, proving to the voter that (with high probability) the receipt was generated correctly. This idea is also seen in Scratch & Vote [2] and Voter Initiated Auditing [3].

3 Voter Experience

Scantegrity II is designed to meet several usability goals. Firstly and most importantly, the procedures for marking and casting a Scantegrity II ballot should deviate minimally from those of a conventional optical scan ballot. Secondly, the presence of a mechanism on the ballot for producing a receipt should not interfere with the marking and casting procedures. Finally, the procedures for producing and checking a receipt should be voluntary.

Here we present a high-level description of the voter experience for the basic system[3]:

Sign-in: A person who is eligible to vote is authenticated at the polling place in accordance with existing procedures governing voter registration. An authenticated voter is issued a single ballot for the purpose of voting and is given a decoder pen.

Normal Voting Routine: The voter enters the voting booth and uses the decoder pen to mark her choices on the ballot in the same manner as one would for a conventional optical scan ballot: a vote is indicated by marking a specially reserved region (referred to as a "bubble") directly beside the printed candidate/choice. The voter casts her ballot, again, in the same manner as one would for a conventional "precinct" optical scan ballot: a poll worker assists the voter in inputting the marked ballot into an optical scanner. The scanner reads the ballot ID and the states of the bubbles marked; it does not read the confirmation codes.

Creating a Receipt: The voter can choose to participate in the voter verification process by creating a receipt of her vote. To create a receipt, the voter manually transcribes the revealed confirmation codes onto the receipt portion of her ballot. Afterwards, the receipt is indelibly marked as "Ballot Voted" by a poll worker (*e.g.*, by a rubber stamp). The voter detaches the receipt from the ballot along a perforation and retains it for future reference. (It is preferable that receipts are detached for all ballots, whether or not the receipts are marked.)

Selecting a Ballot to Audit: During sign-in, a voter can optionally request an additional ballot for the purpose of auditing the correctness of printing. In this situation she is offered two ballots from which she

[3]Note that the voting procedures for the "enhanced" system presented in Section 7 differ slightly from those of this basic system.

arbitrarily selects one to audit. This ballot is indelibly marked as "Audit Ballot" by the poll official (*e.g.*, by a rubber stamp) to unambiguously void its inclusion in the tally. The voter or the poll worker then uses a decoder pen to reveal all the codes on the ballot. The voter is allowed to leave the polling place with this audited ballot for later use in the voter verification step described in Section 4.9.

Exceptions to the Normal Voting Routine: In precinct-based optical scan environments, improperly marked ballots will be immediately rejected by the scanning device. If the local governing procedure allows the voter an additional attempt, the current ballot must first be invalidated before a new one is issued. The receipt (which includes the ballot ID) is detached and stamped as "Spoiled Ballot" and given to the voter. In order to protect voter privacy, the ballot itself should be destroyed in a verifiable manner (*e.g.*, shredded).

Post-Election Voter Verification: During a previously announced time period after the conclusion of polling, the voter may check on the election website that election officials have correctly posted the revealed confirmation codes corresponding to her ballot ID. She may also check that her audit ballot is printed correctly (see Section 4.9). Also, any voter or interested party may check the correctness of the final tally posted on the election web site.

4 The Scantegrity II System

We now describe the details of the basic Scantegrity II system. We consider an election with a single contest for concreteness but without loss of generality. Let B be the number of ballots printed for the election and let N be the number of candidates. B should be at least double the size of the voting population to allow for audited and also spoilt ballots. There is a canonical ordering of the N candidates, but ballots may optionally be printed with varying "ballot rotations".

4.1 The Ballot

The Scantegrity II ballot consists of a *voting portion* and a *receipt portion*, each printed with the ballot's unique ID number. The voting portion of the ballot includes a list of candidate names with an optical mark recognition field, referred to as a *bubble*, beside each candidate name. Each bubble contains a sequence of randomly generated alphanumeric characters, referred to as a *confirmation code*, printed in *invisible ink*. Figure 1 (left) shows the confirmation codes on a sample ballot image file for the printer. Before a ballot is marked, none of the confirmation codes are visible, as shown in Figure 1 (middle). To indicate a vote for a candidate, the voter marks the bubble for that candidate with a special *decoder pen*. Marking a bubble with the decoder pen reveals the confirmation code printed inside the bubble and simultaneously leaves a dark mark that is recognizable by an optical scanner, as shown in Figure 1 (right). The optical scanner uses standard dark-mark detection to count a vote for the candidate exactly when there is a dark mark in the corresponding bubble.

The receipt portion of the ballot contains space for the voter to optionally note the confirmation codes revealed by her selections. For convenience, the receipt portion may contain the list of contest titles with space next to each title for the voter to fill in the code. The receipt portion is located across the bottom of the ballot (so that its perforated edge does not interfere with scanner feeding) and is easily detachable by the voter along a perforation, as shown in Figure 1 (right).

Each ballot has a distinct ballot ID which uniquely identifies the election and the ballot within that election. An example ballot ID might be "ST-2008-11-04-123456789." For simplicity, in the following examples we use 4-digit ballot IDs such as "0001." The ballot ID is printed on both the voting portion and the receipt portion of the ballot. On the voting portion, the ballot ID may appear in the form of pre-filled bubbles for each character of the ID so that the optical scanner can read the ballot ID without using optical character recognition (OCR). Figure 1 (right) shows the ballot ID printed on the voting portion and the receipt portion of the ballot. (The pre-filled bubbles for the ballot ID are not shown.)

4.2 Confirmation Codes

We require the following properties of the confirmation codes:

- The confirmation codes are unique within each contest on each ballot.
- The confirmation code for each candidate within each contest on each ballot is uniformly-pseudorandomly and independently selected from the set of possible codes (which may be restricted to eliminate such things as characters that are commonly confused).
- The confirmation code corresponding to a particular candidate on a particular ballot is secret and unknown to the voter until the voter indicates a vote (*i.e.*, marks the bubble) for that candidate.

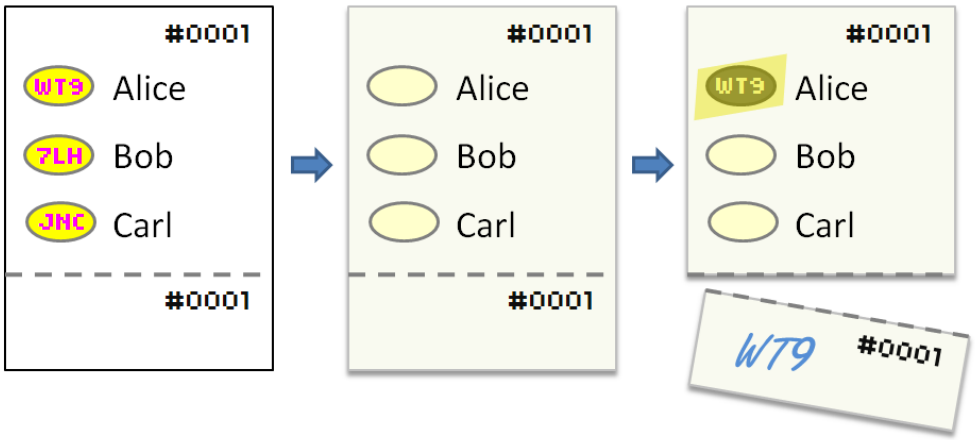

Figure 1: **Example Ballot.** *Left*: Printable ballot image with invisible regions specified by a false-color mapping (magenta and yellow). *Middle*: Printed paper ballot. The confirmation codes, printed in invisible ink, are initially not visible. *Right*: Marked paper ballot and receipt. Marking a bubble with the decoder pen causes the confirmation code to become visible.

Ballot ID	Alice	Bob	Carl
0001	WT9	7LH	JNC
0002	KMT	TC3	J3K
0003	CH7	3TW	9JH
0004	WJL	KWK	H7T
0005	M39	LTM	HNN

Table **P**

Ballot ID			
0001	7LH	WT9	JNC
0002	J3K	TC3	KMT
0003	9JH	CH7	3TW
0004	KWK	H7T	WJL
0005	M39	HNN	LTM

Table **Q**

Flag	**Q**-Pointer	**S**-Pointer
	(0005, 1)	(2, 1)
	(0003, 3)	(4, 2)
	(0002, 1)	(4, 3)
	(0001, 3)	(3, 3)
	(0001, 2)	(4, 1)
	(0005, 3)	(3, 2)
	(0004, 2)	(5, 3)
	(0003, 1)	(2, 3)
	(0004, 3)	(3, 1)
	(0002, 3)	(1, 1)
	(0001, 1)	(2, 2)
	(0002, 2)	(5, 2)
	(0004, 1)	(1, 2)
	(0003, 2)	(5, 1)
	(0005, 2)	(1, 3)

Table **R**

Alice	Bob	Carl

Table **S**

Figure 2: Tables **P**, **Q**, **R**, and **S** as generated by the election officials before election day. Table **P** is kept private. The publicly published versions of tables **Q**, **R**, and **S** contain commitments to the information shown above. For example, a vote for Carl on ballot 0002 would reveal the confirmation code J3K. The corresponding row of table **R** is row 3, which points to position (0002,1) in table **Q** and to position (4,3) in table **S**. Position (4,3) in table **S** corresponds to a vote for Carl, since it is in column 3.

4.3 Election Preparation

Prior to the election, the election officials secret-share a seed to a pseudorandom number generator (PRNG) [26], and election officials then input their shares to a trusted workstation (as described in [14]) to jointly generate $B \cdot N$ pseudorandom confirmation codes and the following tables:

P : A table containing the confirmation codes in the order in which they were generated by the PRNG. Table **P** specifies the correspondence between confirmation codes and candidates on each ballot. Row i corresponds to ballot i and column j corresponds to candidate j, so that the confirmation code in position (i, j) is printed on ballot i within the bubble for candidate j. Table **P** is never published and is used to generate table **Q**.

Q : A table in which the confirmation codes in each row of **P** have been pseudorandomly permuted. Thus, row i corresponds to ballot i, but each column does not correspond to a fixed candidate. The election officials commit to each confirmation code in table **Q** and publish these commitments on the election website.

R : A table in which each row i corresponds to an underlying confirmation code from **Q**. Each row contains a flag, which will be raised in the post-election posting phase if a vote is made for the underlying confirmation code, and two pseudorandom pointers — a "**Q**-pointer" specifying the position of a confirmation code in table **Q** and an "**S**-pointer" specifying the position of the same confirmation code in table **S** (described below). The election officials generate these pointers using the PRNG, commit to each **Q**-pointer and **S**-pointer, and publish these commitments on the election website. Essentially, table **R** provides two random shuffles of the confirmation codes and will be used in the audit process via randomized partial checking [17].

S : A table in which each element corresponds to an underlying confirmation code. Each element is a flag, which will be raised if a vote is made for the underlying confirmation code. Each column j contains the confirmation codes for candidate j. Table **S** (initially empty) is published on the election website.

Because the **Q**- and **S**-pointers in table **R** are pseudorandomly generated, an element in table **Q** can map to any element of table **R**, which can map to any element of table **S** in the column for the corresponding candidate.

As a working example, consider an election with $N = 3$ candidates, Alice, Bob, and Carl, and $B = 5$ ballots. Figure 2 shows how the private tables **P**, **Q**, **R**, and **S** generated by the election officials might look before election day.

4.4 Voting

After checking in at the polling site, the voter asks for either one or two Scantegrity II ballots and a decoder pen. If the voter asks for two ballots, she chooses one arbitrarily to audit (see Section 4.9) and votes on the other one. Inside the booth, to vote for a candidate, the voter marks the bubble for that candidate using the decoder pen, simultaneously revealing the confirmation code printed inside the bubble and leaving a dark mark that is recognizable by the optical scanner.

A voter who is interested in verifying that her vote is collected as cast may transcribe the revealed confirmation codes onto the receipt portion of the ballot. The voter detaches the receipt portion of the ballot along a perforation.

As with conventional optical scan, the voter inserts her ballot into the optical scanner, which checks that the voter has not overvoted and that there are no stray marks. If there is an error, the scanner may reject the ballot and the voter can then either return to the booth to make additional marks or obtain a new ballot to vote on. Otherwise, the scanner accepts the ballot and records the voter's choices, along with the ballot ID.

After the voter casts her ballot, a poll worker stamps the voter's detached receipt as "Ballot Voted".

4.5 Posting and Tallying

After the close of polls, the election officials publish a list of all the voters who voted and the tally given by the optical scanners. The electronic ballot images from the scanner and table **P** are used to translate the votes into the confirmation codes which were revealed on the cast ballots. The election officials open the commitments in table **Q** to the confirmation codes that have been revealed to voters and flag the entries in tables **R** and **S** corresponding to those codes. Anyone can now compute the number of votes for each candidate as the sum of the number of flagged entries in the candidate's column in table **S**. These tallies are checked against those reported by the optical scanners.

For each row of **R**, election officials open either the **Q**-pointer or the **S**-pointer, depending on a true random, publicly verifiable coin flip [11]. This information will be used to audit the election website via randomized partial checking (see Section 4.8).

For those ballots that were spoiled for audit, election officials open the commitments to all the information associated with the ballot, *i.e.*, the confirmation codes in **Q**

and the **Q**- and **S**-pointers in **R**.

Suppose ballots 0001 and 0003 are votes for Alice, ballot 0002 is a vote for Carl, ballot 0005 is a vote for Bob, and ballot 0004 is spoiled for audit (as described in Section 4.9). Figure 3 shows how the publicly published **Q**, **R**, and **S** tables would look at the end of election day.

4.6 Checking Receipts

Once the results are posted, a voter who has made a receipt of her confirmation codes can go to the election website and look up her ballot ID. She checks that, in the row of **Q** corresponding to her ballot ID, all and only her confirmation codes appear. Anyone can check that the opened commitments match the confirmation codes on the election website. If any of the confirmation codes from the voter's receipt does not appear posted in **Q**, the voter should file a dispute.

4.7 Dispute Resolution

To file a dispute, the voter must be on the registration list as being eligible to vote or as having cast a ballot, and only one dispute may be filed per voter. (If the voter wishes to allow a third party organization to check her receipt, she might sign over the right to her dispute.) The voter provides the ballot ID and a claim of what the correct confirmation code(s) should be.

Disputes may arise under four general circumstances. They could be (i) the result of the voter making a transcription error, (ii) a mischievous voter attempting to call into question the legitimacy of the election, (iii) an error by the scanner or in the election data, or (iv) evidence of fraudulent behavior.

The election officials would like to distinguish the latter two cases from the former two, as the latter two cases undermine the integrity of the election and should trigger further investigation. Since voters do not see the confirmation codes of the unrevealed candidates on the ballot, the probability that a code that a voter claims to have revealed is one of the other codes committed to for her ballot is very small if the voter made a transcription error or is merely guessing. The election officials can publicly open the commitments to the other confirmation codes for the claimant's ballot and eliminate from consideration any disputes for which none of the opened codes matches the claimed code. We consider the remaining disputes to be "plausible discrepancies."

Plausible discrepancies could still be the product of cases (i) or (ii); however, the statistically expected number of them arising from random guessing is small and can be quantified. The election officials should set up a statistical trigger, based on various election parameters, such that if the number of plausible discrepancies exceeds some threshold, some suitable recourse is instigated.

Here we consider one possible statistical trigger. Consider a single disputed race. Let N be the number of candidates, C be the code space, D be the number of disputes filed, and G be number of plausible discrepancies filed. Let p be the probability of randomly guessing a code that constitutes a plausible discrepancy. Then $\mu = Dp$ is the expected value of G if all filed disputes are random guesses. We set the trigger τ such that the probability of obtaining at least τ plausible discrepancies if all filed disputes are random guesses is less than 1%. We can use the following bound on the right tail of the binomial distribution [12]. For any $r > \mu$, $\Pr[G - \mu \geq r] \leq (\mu e/r)^r$. For example, for $N = 5$ candidates, $C = 8000$ possible codes, and $D = 1000$ disputes filed, assuming no scanning error, $p = 4/7999 = 0.0005$ and $\mu = 1000 \cdot 0.0005 = 0.5$. Using $r = 4.5$ we get $\Pr[G \geq 5] \leq (0.5e/4.5)^{4.5} = 0.0046 < 0.01$, so we can set $\tau = 5$. If at least 5 out of the 1000 disputes filed are plausible discrepancies, then an investigation should be instigated. To allow for up to some acceptable rate s of scanning error, we can incorporate s into the probability p of guessing a correct code and compute the statistical trigger as above with the new value of p.

4.8 Checking the Tally

We audit the election website using randomized partial checking (RPC) [17]. For each element in **R**, the election official is asked to open either the **Q**- or the **S**-pointer, depending on a pseudorandom publicly verifiable coin flip [11].

Any interested party can check that the commitments are correct, that each revealed **Q**-pointer in table **R** either connects a revealed code in table **Q** to a flagged element in table **R** or connects a hidden code to an unflagged element, and that each revealed **S**-pointer in table **R** either connects a flagged element in table **R** to a flagged element in table **S** or connects an unflagged element to an unflagged element. Essentially, the audit checks that flags are mapped unchanged from table **Q** through table **R** to table **S**.

Finally, any interested party can check that the tally for each candidate is correctly computed as the total number of flagged entries in that candidate's column in table **S**.

4.9 Auditing Ballots

Auditing of ballots may be done at any point during the election process. Candidates should pre-audit a random sample of the ballots prior to election day. On election day, voters may obtain two ballots and audit a random

Ballot ID			
0001		WT9	
0002	J3K		
0003		CH7	
0004	KWK	H7T	WJL
0005			LTM

Table **Q**

Flag	**Q**-Pointer	**S**-Pointer
		(2,1)
	(0003,3)	
✓		(4,3)
		(3,3)
✓	(0001,2)	
✓	(0005,3)	
	(0004,2)	(5,3)
		(2,3)
	(0004,3)	(3,1)
	(0002,3)	
	(0001,1)	
	(0002,2)	
	(0004,1)	(1,2)
✓		(5,1)
	(0005,2)	

Table **R**

Alice	Bob	Carl
	✓	
✓		✓
✓		

Table **S**

Figure 3: Tables **Q**, **R**, and **S** as published after the close of the election. For each revealed confirmation code in table **Q**, the row corresponding to that code in table **R** and the element corresponding to that code in table **S** have been flagged. For each row of table **R**, either the commitment to the **Q**-pointer or the commitment to the **S**-pointer has been opened and published. Note that the revealed confirmation codes in table **Q** (other than those for ballot 0004, which is a ballot chosen for audit), the rows flagged in table **R**, and the flags in table **S** are in one-to-one correspondence.

one and vote with the other. At the end of the day, any remaining blank ballots may also be audited.

When a ballot is selected to be audited, a pollworker stamps the ballot as "Audit Ballot" on both the voting portion and the receipt portion and all of the confirmation codes on the ballot are revealed using a decoder pen. We will refer to the person checking an audited ballot as the auditor. The auditor checks that the confirmation codes are unique within each race. After election officials open the commitments to the confirmation codes in **Q** and the **Q**- and **S**-pointers for those confirmation codes, the auditor checks that the published confirmation codes match those revealed on the paper ballot, and that the published pointers map each confirmation code in **Q** to the appropriate candidate column in **S**.

5 Security and Privacy Discussion

Scantegrity II provides improved integrity properties over the underlying optical scan system and only minimally impacts the privacy and ballot secrecy properties. We provide a high-level sketch of several common attacks and the countermeasures Scantegrity II utilizes in thwarting them.

In our threat model, an adversary could be any party, including a voter, pollworker, or election official. The goal of the adversary is to attack the integrity of the election or exert undue influence over a voter. We do not consider attack vectors applicable to all voting systems, such as denial of service attacks, given that these attacks and countermeasures are not unique to Scantegrity II. Furthermore, we note that the security of Scantegrity II is equivalent in many regards to that of Punchscan and the original Scantegrity, for which security analyses are already present in academic literature [14, 8]. We split our discussion of threats into those that affect integrity and those that affect privacy and ballot secrecy.

5.1 Integrity

Integrity of an election refers to the inability of an adversary to modify votes undetectably. Compared to conventional optical scan, Scantegrity II provides improved ability to detect an attacker attempting to affect the integrity of an election by modifying ballots, creating fraudulent ballots, or attempting to change the tally on the election website.

5.1.1 Assumptions

We make the following assumptions for the integrity of the system:

1. A minimally sufficient number of voters will check their receipts and audit ballots.

2. The system uses a cryptographic commitment function that is computationally binding.

3. Observers monitor the election website and can immediately save and detect changes.

4. An effective voter registration system is used.

5. Auditors will not collude with the election officials.

5.1.2 Adding or Deleting Votes

In order to add ballots to the tally, an adversary would need to add names to the list of voters who voted, in addition to changing the number of voters reported by polling sites. If an adversary attempts to add a significant number of ballots this should be easily detected by someone.

An adversary might attempt to delete ballots from the bulletin board by marking voted ballots as spoiled for audit and publishing all of the information associated with those ballots. Then he runs the risk of the voters of those ballots checking the election website and protesting that their ballots were incorrectly posted as spoiled for audit, providing their stamped receipts as proof.

If an adversary adds more marks to a ballot after it has been cast, it can be detected by public observation of the election website if the additional marks turned a valid ballot into an overvoted one. Some optical scan systems prevent this attack because they do not accept overvoted ballots, but the adversary could add a mark to a ballot which was originally an undervote. This can be detected by the voter of that ballot, but the voter has no proof that the mark was added later. One way to address this is to always include a "None of the above" option on each race, which voters could optionally mark to submit an undervote.

5.1.3 Fraudulent Ballots

There are several types of fraudulent ballots that the election officials could create or that an adversary could substitute for properly constructed ballots:

- There are repeated confirmation codes within a single race on a ballot. This means that even if the correct confirmation code is posted, a voter cannot be assured that her vote was collected as cast because multiple candidates may share the same confirmation code.

- The confirmation codes on the ballot are printed in a different order from that specified by the **P** table, or the **Q**- and **S**-pointers for the codes on the ballot do not map the codes to the correct candidate columns in **S**. This means that even if the correct confirmation code is posted, a voter cannot be assured that her vote was collected as cast because the confirmation code may be for a different candidate.

- The ballot may be completely invalid, *e.g.*, the confirmation codes on the ballot do not match the confirmation codes committed to, in any order.

In all of these cases, the fraudulent ballots can be detected by the cut-and-choose auditing of the ballots. If a random half of the ballots are audited, no more than a few fraudulent ballots should go undetected.

More precisely, if there are B total ballots, F fraudulent ballots, and A ballots are audited, then the probability that the adversary will go undetected is $\binom{B-F}{A}/\binom{B}{A}$, which can be straightforwardly shown to be upper bounded by $\min[(1-A/B)^F, (1-F/B)^A]$.

5.1.4 Modifying the Tally

An adversary may try to change the vote tally by flipping flags in **Q**, **R**, or **S**, so that counting the flags for each candidate in table **S** gives an incorrect tally. This can be detected by the randomized partial checking of the **Q**- and **S**-pointers in **R**. The probability of an adversary modifying k pointers going undetected is $1/2^k$.

5.2 Privacy

Privacy refers to the inability of an attacker to link a voter's candidate choices to the voter. Modeling privacy in a voting system requires strong assumptions due to the wide variety of physical attacks that could violate voter privacy in any voting system. Given our assumptions, Scantegrity II offers the same privacy protection as for conventional optical scan systems. However, privacy concerns arise if these assumptions are relaxed. The enhancements presented in Section 7 address these privacy concerns.

5.2.1 Assumptions

We make the following assumptions for the privacy of the system.

1. There are no recording devices in the polling booth.

2. Voters cannot read the confirmation codes unless they mark the corresponding bubble.

3. The system uses a cryptographic commitment function that is computationally hiding [20].

4. Procedural mechanisms including strict management of access to ballot boxes are enforced.

5. Election officials use a special trusted computer workstation (as described in [14]) to enforce the privacy of the tables of confirmation codes.

5.2.2 Randomized Partial Checking

For each code, either the **Q**-pointer or the **S**-pointer is revealed. Suppose the **Q**-pointer for a particular code is revealed. Then the index of the code in table **S** is one of those not pointed to by a revealed **S**-pointer. Similarly, if the **Q**-pointer for a code is not revealed, then the **S**-pointer is revealed, so the index of the code in table **S** is one of those pointed to by a revealed **S**-pointer. Thus, the randomized partial checking of the pointers hides each voter's vote equally among a randomly selected half of the votes.

5.2.3 Receipts

Because the confirmation codes are randomly generated, randomly assigned to candidates, and independent across different ballots, and the commitment function is computationally hiding, a voter's receipt of confirmation codes does not reveal how she voted. As a result, an attacker cannot coerce a voter to vote one way or another without prior knowledge of the ballot she receives and the codes on that ballot.

If the generation of the confirmation codes or the secrecy of the private information in the tables is compromised, an adversary could violate a voter's privacy. If an adversary obtains access to cast ballot forms and obtains the ballot ID used by a voter, he will be able to determine that voter's choices. To address these threats, we make the assumptions that no recording devices are present in the polling booth and chain-of-custody procedures are enforced to ensure that an adversary does not have access to the ballot boxes. Further countermeasures are possible, for example, ensuring that the ballot ID on the cast portion of the ballots is difficult to extract. One possibility is to encrypt the ballot ID or make it not human-readable.

6 Invisible Ink Printing Process

Both the ballot and confirmation codes are printed before voting takes place. Three types of inks are employed for this purpose. Conventional (*e.g.*, black) ink is used for primary printing of candidate names, instructions, other text, etc. For the purposes of printing confirmation codes, two specialty inks are employed: a reactive invisible ink and an unreactive color-matched dummy ink.

The term "invisible ink" refers to an ink with the following properties:

- In its initial chemical state, its pigmentation is of low visual contrast to the print medium (*i.e.*, paper).
- When combined with a developer ink, an irreversible reaction occurs, changing the pigmentation of the ink to be of high visual contrast to the print medium.
- In its activated state, it must be sufficiently pigmented to be clearly visible to an optical scanner and the human eye.

However, the term "invisible ink" is a misnomer in the sense that any liquid (though not necessarily pigmented) will subtly change the reflective characteristics of the print medium through the printing process, meaning the presence of letters printed in invisible ink also will still potentially be visually perceptible. For this reason we employ a dummy ink with the following properties:

- It is color-matched to the initial pigmentation characteristics of the invisible ink such that the inks are visually *indistinguishable*.
- Contact with the developer ink will produce negligible pigmentation change so as to be in high visual contrast to the activated ink such that the inks become readily visually *distinguishable*.

We define a matrix of pixels for the purposes of printing the confirmation codes inside the optical scan bubbles. Pixels that form the "foreground" confirmation code characters are printed in one ink, while the remaining "background" pixels are printed in the other ink. In order to produce the darkest average color density of a marked optical scan bubble (and therefore most visible to the scanner), the background pixels are printed with the invisible ink, and the foreground pixels are printed in the unreactive dummy ink. Therefore when marked by the decoder pen, the bubble will contain lightly colored letters against a dark background.

The developer ink is incorporated into a felt-tipped "highlighter" style marker casing along with a basic yellow pigment to provide voters with visual feedback as to where they have marked. We have implemented the invisible ink printing process using an Epson C88+ color inkjet printer in which we replace the manufacturer-supplied yellow and magenta inks with the reactive and non-reactive inks respectively. Electronic files used for ballot printing are prepared in this false-color mapping whereby yellow and magenta colored objects in the digital file will print in reactive and non-reactive inks respectively. This process is depicted in Figure 4.

7 Enhancements to the Basic System

In this section, we suggest ways to improve the basic Scantegrity II system described thus far. One enhancement allows all information on the ballot to be humanly-visible only within a short window of time. Another

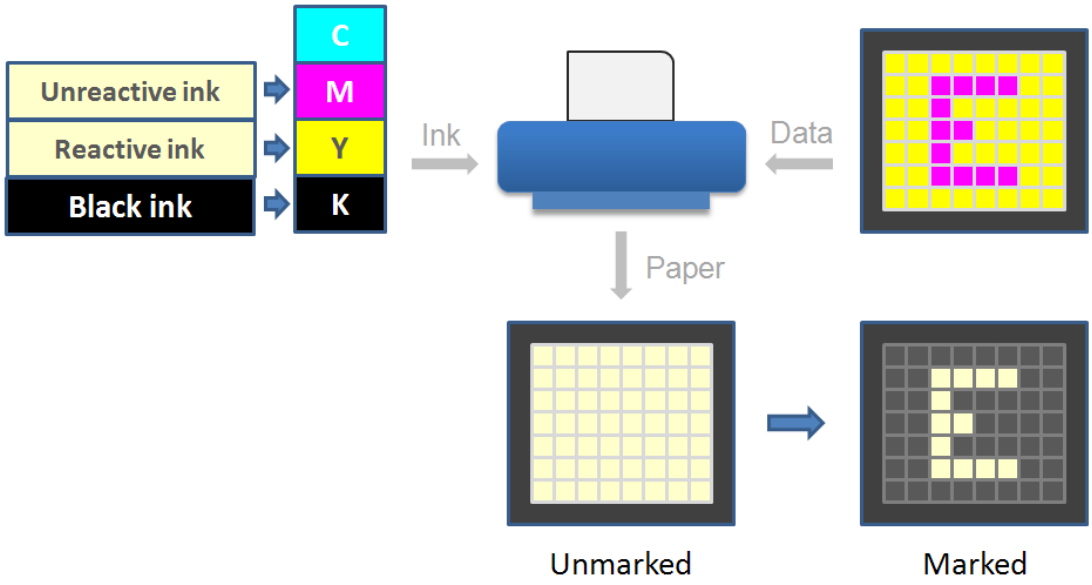

Figure 4: **Invisible Ink Printing Process.**

improvement allows information seen by the voter to be sufficient for the voter to complain, thereby reducing the efficacy of attacks in which physical receipt may be obtained from voters.

In the basic system, uniquely identifiable codes are left on the paper ballots at the conclusion of the election, making manual recounts potentially useful in improper influence schemes, such as vote buying and coercion. Even though manual recounts are a weaker integrity guarantee than that offered by the Scantegrity II audit, it is preferable to be able to allow both. Our first enhancement, designed to permit recounts, utilizes a "slow-reacting" invisible ink—created by diluting the concentration of the reactants in the normal invisible ink or introducing a chemical inhibitor. When brought into contact with the developer ink, this ink has a much lower reaction rate—on the order of tens of minutes instead of fractions of a second. By replacing the dummy ink on the foreground pixels of each confirmation code on a ballot with slow-reacting ink, a bubble undergoes an immediate reaction upon marking, making the codes visible to the voter. A slow reaction will, however, gradually (*e.g.,* over 15 minutes) begin to fade the foreground pixels of the confirmation code to an indistinguishable color, filling in the bubble completely and hiding the confirmation code. The ballots can then be re-scanned and/or manually counted or recounted, as often as desired, without revealing confirmation codes to the persons involved.

When a ballot is audited, the confirmation codes can be revealed using one of the decoder pens used for voting (to audit the effectiveness of the pen) or taken home by the voter and developed if she has access to the ink. To prevent the confirmation codes from fading, the voter can choose to apply a fixative, by spraying or sponging her ballot, that completely inhibits the slow-reacting ink and preserves the codes.

Disputes in the basic system require physical possession of a stamped receipt. The second enhancement allows for informational disputes, where a voter need only know her confirmation codes in order to prove that her ballot is misrepresented online. Further, to prevent the threat of voted ballots being turned into audited or spoiled ballots without stamped receipts, two additional "authenticated ballot status" codes are added to the ballot. These status codes are the same size as the confirmation codes, are committed to prior to the election, printed on the ballot in slow-reacting ink, and are individually detachable (*e.g.,* using a perforation). If a ballot is chosen for a print audit by the voter, one of these codes is developed by the poll worker and the other detached and destroyed—which one is the choice of the voter. If a ballot is chosen to be voted on, both codes are retained on the ballot, detached prior to feeding the ballot into the optical scanner, and developed by the poll worker after the ballot is successfully cast. Within a few minutes, the voter is able to read the codes. If the ballot is spoiled (*e.g.,* the voter overvoted and is issued another ballot), the chit is returned by the voter and checked as matching the ballot being spoiled and then both codes are shredded without being developed. If a voter later finds that

her cast ballot has been misrepresented as an audited or spoiled ballot on the website, knowledge of both of the codes is necessary and sufficient for proving the ballot was voted, indicating malfeasance or error. Similarly, if a voter discovers that her audited vote has been misrepresented as a voted or spoiled ballot, knowledge of one code is sufficient to prove that the ballot was not spoiled and knowledge of all the confirmation codes is sufficient to prove that the ballot was not voted on.[4] In order to dispute an incorrect confirmation code that appears online, the voter need only know a proper code and does not need to actually have the physical receipt (stamped or otherwise). Given that voters are provided adequate time in the booth to memorize or copy the codes, obtaining a voter's stamped receipt after the election is not sufficient to prevent the voter from detecting an error on the election website and filing a successful dispute.

Since no physical evidence is required for a dispute, the dispute resolution process does not have to occur in person and the voter can file an online petition instead (optionally anonymously). This may be advantageous in situations where the voter has reason to fear retribution for reporting errors. To prevent the system from being what may be called "flooded" with spurious complaints, sensible precautions can be taken to eliminate blatantly fraudulent claims, including CAPTCHAs and limits on complaints per IP address. If dispute flooding occurs, however, the resolution can move to a second phase where candidates or political parties are allocated a fixed number of complaints per flooded ballot and can then choose the most plausible complaints as they prefer.

These improvements do not interfere with Scantegrity II's ability to be used in central-scan environments. As the scanner does not need to record the confirmation codes, since the positions marked are sufficient to reconstruct the codes, ballots may be scanned after the slow-reacting ink has hidden the confirmation codes. This is useful for precincts that lack scanning equipment, as is the case in many developing democracies. Scantegrity II can also augment hand-counted elections by allowing the ballot-counting to proceed as usual, and then providing the ability for the ballots to be centrally scanned after the election to ensure end-to-end integrity.

8 Conclusions

Scantegrity II provides voters with a familiar optical scan voting experience as well as an option to verify that their ballots "make it, as intended, all the way." This end-to-end verification capability enables election officials to provide the highest level of assurance that outcomes are correct and in a way that is visible to voters. The system also improves on its predecessor by removing the need to recover paper ballots in resolving disputes about the information posted on the election website and allows for manual recount.

While further testing and development are planned, Scantegrity II holds promise of being the first end-to-end voting system to come into use in public sector elections.

[4] An additional two codes could be added in a similar fashion to prove that spoiled ballots are not misrepresented as voted or audited ballots; however, mere detection of this threat, in lieu of prevention, is a marginal improvement as previously outlined.

9 Acknowledgments

We are grateful to Russell Fink for reviewing several versions of the paper, and the Ottawa Canada Linux Users Group for test-driving our system and providing useful feedback.

References

[1] Ben Adida and C. Andrew Neff. Ballot casting assurance. In *EVT'06: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop*, page 7, Berkeley, CA, USA, 2006. USENIX Association.

[2] Ben Adida and Ronald L. Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In *WPES '06: Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, pages 29–40, New York, NY, USA, 2006. ACM Press.

[3] Josh Benaloh. Ballot casting assurance via voter-initiated poll station auditing. In *Preproceedings of the 2007 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT 2007)*, August 2007.

[4] Giles Brassard, David L. Chaum, and Claude Crepeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37:156–189, 1988.

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

[6] David Chaum. Secret-ballot systems with voter-verifiable integrity. United States Patent and Trademark Office, 7,210,617, January 2003.

[7] David Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security and Privacy*, 02(1):38–47, 2004.

[8] David Chaum, Aleksander Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, Alan T. Sherman, and Poorvi Vora. Scantegrity: End-to-end voter verifiable optical-scan voting. In *IEEE Security and Privacy*, volume May/June, 2008.

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

[10] David Chaum, Jeroen van de Graaf, Peter Y. A. Ryan, and Poorvi L. Vora. High integrity elections. Cryptology ePrint Archive, Report 2007/270, 2007. `http://eprint.iacr.org/`.

[11] Jeremy Clark, Aleksander Essex, and Carlisle Adams. Secure and observable auditing of electronic voting systems using stock indices. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2007.

[12] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press and Mc-Graw Hill, second edition, 2001.

[13] Aleks Essex, Jeremy Clark, Richard Carback, and Stefan Popoveniuc. Punchscan in practice: An E2E election case study. In *Preproceedings of the 2007 IAVoSS Workshop on Trustworthy Elections (WOTE 2007)*, June 2007.

[14] Aleks Essex, Jeremy Clark, Richard Carback, and Stefan Popoveniuc. The Punchscan voting system: VoComp competition submission. In *Proceedings of the First University Voting Systems Competition (VoComp)*, 2007.

[15] Kevin Fisher. Punchscan: Security analysis of a high integrity voting System. Master's thesis, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, December 2006.

[16] Kevin Fisher, Richard Carback, and Alan T. Sherman. Punchscan: Introduction and system definition of a high-integrity election system. In *Preproceedings of the 2006 IAVoSS Workshop on Trustworthy Elections*, Robinson College, Cambridge, United Kingdom, 2006. International Association for Voting System Sciences.

[17] Markus Jakobsson, Ari Juels, and Ronald Rivest. Making mix nets robust for electronic voting by randomized partial checking. In *Proceedings of the 11th USENIX Security Symposium*, pages 339–353, Berkeley, CA, USA, 2002. USENIX Association.

[18] John Kelsey, Andrew Regenscheid, Tal Moran, and David Chaum. Hacking paper: some random attacks on paper-based E2E systems. 2007.

[19] David Lundin. Component based electronic voting systems. In *Preproceedings of the IAVoSS Workshop on Trustworthy Elections (WOTE 2007)*, 2007.

[20] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 129–140, London, UK, 1992. Springer-Verlag.

[21] Stefan Popoveniuc, Jeremy Clark, Richard Carback, and Aleks Essex. Securing optical-scan voting. In *Proceedings of Frontiers of Electronic Voting (FEV 2007)*, 2007.

[22] Stefan Popoveniuc and Ben Hosp. An introduction to punchscan. In *Preproceedings of the 2006 IAVoSS Workshop on Trustworthy Elections*, Robinson College, Cambridge, United Kingdom, 2006. International Association for Voting System Sciences.

[23] Stefan Popoveniuc and Jonathon Stanton. Undervote and pattern voting: Vulnerability and a mitigation technique. In *Preproceedings of the 2007 IAVoSS Workshop on Trustworthy Elections (WOTE 2007)*, June 2007.

[24] Stefan Popoveniuc and Poorvi Vora. A framework for secure electronic voting. In *Preproceedings of the IAVoSS Workshop on Trustworthy Elections (WOTE 2008)*, 2008.

[25] Peter Y. A. Ryan and Thea Peacock. Prêt à Voter: A systems perspective. `http://www.cs.ncl.ac.uk/research/pubs/trs/papers/929.pdf`, September 2005.

[26] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.