ABSTRACT

Title of Dissertation: Engineering Practical End-to-End Verifiable Voting Systems

Richard T. Carback III, Doctor of Philosophy, 2010

Dissertationdirected by: Alan T. Sherman, Associate Professor
Department of Computer Science and
Electrical Engineering

We designed, built, tested, and fielded a vote counting system called Scantegrity. Scantegrity is part of a new class of end-to-end (E2E) verifiable voting systems.

E2E verifiable systems are designed to solve chain of custody problems in elections by providing a privacy-preserving receipt to each voter. The voter can use the receipt to check a public record and verify that his or her ballot is counted without revealing the selected candidate. The public record gives election officials the ability to provide a strong, universally-accessible audit of the results, enabling transparent, verifiable elections that maintain privacy expectations. E2E systems offer radical improvements to integrity and transparency of election systems, and the adoption of E2E systems in public-sector elections can improve outcome integrity.

In our design, we carefully considered the balance between usability and security issues, and we discuss the changes we made to implement the system. We examined the implementation through the results of a practical test of Scantegrity in a mock election in April 2009, which evaluated the system's performance and surveyed the election participants about their experience with the system.

We describe a number of changes we made to the system as a result of this test. For example, Scantegrity required better printing technology and a tally reconciliation system. We evaluated the modified system a second time by fielding it in the Takoma Park, Maryland, November 2009 municipal election, where we were able to survey voters and observe events throughout election day. In addition to examining the performance of the system during election day, we analyzed the survey results and found that most voters felt positively about the system despite some problems when taking advantage of the new features.

We suggest further improvement to the usability of Scantegrity by proposing and analyzing the addition of an automatic receipt printer in different configurations. To design the receipt printer, we took advantage of protections provided by the Trusted Computing platform that improve the reliability and robustness of the component. The final system automatically provides each voter a privacy-preserving receipt that can be used to verify each ballot has been counted properly.

Engineering Practical End-to-End Verifiable Voting Systems

by

Richard T. Carback III

Dissertation submitted to the Faculty of the Graduate School
of the University of Maryland, Baltimore County in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

To my wife, Alice, who continues to provide endless support and patience.

ACKNOWLEDGMENTS

First, I am grateful to the many people who made the studies of Scantegrity possible, especially Anne Sergeant (Chair, Takoma Park Board of Elections), other members of the Board, Jessie Carpenter (City Clerk), and the Takoma Park voters. I would also like to thank the auditors—Lillie Coney, Ben Adida, and Filip Zagórski—who independently created election authentication software and performed other auditing activities. Not only did they verify our work, but they helped spread the word about end-to-end (E2E) verifiable voting.

Alan Sherman was already my master's thesis advisor, and he continues to be a source of helpful advice and feedback. It has been a pleasure to work with him for all of these years, and I will miss the interaction.

Poorvi Vora has been a close mentor and friend. She has provided advice, encouragement, and new books to satisfy my curiosity. She has also been a very important guide and contributor during our research calls, often playing the voice of reason and practicality.

David Chaum, who is behind a substantial number of cryptographic breakthroughs, has been a long-distance mentor for the last five years. His presentation on the interface he invented for Punchscan originally inspired my interest in voting, and he is an endless source of new ideas, including the Scantegrity system.

Jeremy Clark, Aleksander Essex, and Stefan Popoveniuc—along with David and I—made up the core research team who made both Punchscan and Scantegrity a reality, and

without them these systems might not have made it past the proposal stage. They are all brilliant and each have specific invaluable strengths, and I hope to work with them on future projects.

Likewise, the newer research team members have had significant impacts on the Scantegrity system. Ronald Rivest, who is my grand advisor and is also behind many cryptographic breakthroughs, and his student, Emily Shen, provided incredible insight into and analysis of the Scantegrity proposals. My programming partners, John Conway and Travis Mayberry, made many improvements to the Scantegrity system, and in turn provided many new ideas and process improvements to the rest of the team.

For the mock and municipal election studies, Paul Herrnson, a social scientist and leader in voting technology analysis, provided critical expertise and advice when conducting our observational studies, and taught me everything I know about survey analysis. Bimal Sinha also contributed his substantial knowledge of applied statistics and helped me make sure that we were on track in our analysis.

During the mock election studies, Lynn Baumeister led the focus groups and offered numerous practical suggestions. Brian Strege and Fahad Alduraibi observed voters. Russell Fink, Douglas Jones, Sharon Laskowski, and Svetlana Lowry provided useful feedback.

During the municipal election, Vivek Relan and Bhushan Sonawane timed voters as they voted and helped assemble the privacy sleeves. Lynn Baumeister interviewed some voters as they left the precinct. Cory Jones provided general assistance and Alex Florescu and Jan Rubio assisted with ink creation.

For the receipt printer design, I wish to acknowledge Ronald Rivest for suggesting the kernel of the image duplicator design, Christina Wnuk-Fink for discussions on the practicality of receipt printing for Scantegrity, and members of the University of Maryland, Baltimore County (UMBC) Cyber Defense Lab and Johns Hopkins University Applied

Physics Laboratory for review of the manuscript. Russell Fink and Sherman were my collaborators when creating the receipt printer designs.

Lindley Ashline proofread and offered editorial assistance for most of the chapters in this dissertation. Esther Haynes offered editorial suggestions on the manuscript for the mock election publication.

During part of this research, I was supported by UMBC with teaching assistantships, and Sherman also provided some summer work on cyber defense lab exercises through the Department of Defense Information Assurance Scholarship Program (DoD IASP). When not supported by UMBC, I thank my current employer, Convergent Technologies, Inc., for being flexible enough to allow me to take the time I needed to complete my research.

Sherman was supported in part by the DoD under IASP grants H98230-08-1-0334 and H98230-09-1-0404. Vora was supported in part by the National Science Foundation under grant CNS 0831149. Jeremy Clark and Aleksander Essex were supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A modern election is a complex, distributed process. A central election authority coordinates the actions of multiple polling sites throughout local jurisdictions. The authority packages and delivers materials and equipment to each polling site, a team of bipartisan election judges—volunteers who take an oath, receive little training, and are paid a token wage—are deployed at each polling site to conduct the election, and, at the end of election day, each polling site reports results back to the central authority that reports these results to the public.

The central authority relies on a protection mechanism known as a *chain of custody* throughout the entire election. For each stage of the election, one or more judges or other officials sign for the previous official and takes custody of the materials and equipment for next stage. Depending on the stage, the signing official may verify tamper-evident security tape seals or perform other actions before taking custody. At the end of an election the central authority must reconcile all equipment and materials from each polling site. A mistake or vulnerability at any point in this chain of custody can corrupt the results of the election and destroy public trust in the government.

A limited number of observers can audit this chain, but *quis custodiet ipsos custodes* (who watches the watchmen)? There are physical limits on what any one observer can verify, and there are often too few observers for a comprehensive audit. More importantly, if there is an issue, an observer may not have any recourse or evidence to justify his claim. When we add automated voting machinery designed to make the voting process easier and more accessible to voters, the observer's job may become impossible if there is no way to verify the software and hardware.

As voters, we are forced to trust officials to implement the chain of custody flawlessly. There is no verification mechanism for a voter in the modern election. We sacrificed the verifiability provided by a show of raised hands in a room or a public posting of each voter's choice for a secret ballot. This sacrifice protects us from coercion, but takes away assurance that our vote is properly captured in an election tally.

In this dissertation, we discuss the construction of practical, high-assurance, end-to-end (E2E) verifiable voting mechanisms that give verifiable elections back to voters.[1] These systems provide a privacy-preserving receipt to each voter. The receipt allows the voter to use her receipt to check that her ballot is included in a public election record. The public record lets election officials provide a strong, universally-accessible audit of the results, enabling transparent, verifiable elections that maintain privacy expectations. Without an E2E-like mechanism, we are unaware of any technology that permits election officials to conduct genuinely verifiable, secure, distributed secret ballot elections.

[1] Also known as *open-audit voting*, the term "end-to-end" is from the 2006 TGDC NIST Proposal [140].

1.1 The Voting Problem

The purpose of an election is to poll the opinion of each voter to make collective decisions. The main requirements are that voters can express their opinions and that they be counted accurately. The concept is simple, and the execution could also be simple. It certainly has been in the past, as collective decisionmaking has existed since ancient times. However, requirements have changed within the past few centuries, and can be attributed to the social needs of a secret ballot and accessibility.

Secret Ballots. Secret ballots allow voters to vote confidentially. The need for secret ballot elections arose in response to widespread intimidation and bribery, and a secret ballot is necessary to gain a true and sincere response from voters.

Accessibility. Each adult citizen, with some exceptions, has a right to vote in elections held in the jurisdiction where he or she keeps primary residence. A voter could be traveling or immobile and need to vote remotely. A voter might be illiterate, a non-native speaker of the dominant language in the area, have physical or mental disabilities, or have some other issue that makes voting difficult. To make the election accessible, we must use procedures and equipment that will accommodate these voters and provide, as much as possible, the same guarantees to all voters.

Secret ballots and accessibility requirements are significant issues, and they are the source for most of the necessary complexity present in modern elections. Ideally, a voting system should be accessible, verifiable, and able to maintain the secret ballot. Each voter should have the ability to verify that her vote is properly captured, and each observer should be able to verify that all the recorded votes are properly counted.

While the advent of computerization in modern elections has vastly increased the accessibility of voting systems, it has done so by compromising security. The operations of current systems are proprietary and opaque to voters and observers. Basic protections such as a redundant paper record are often missing. Creating election systems that are accessible and secure is difficult, and we believe that election systems should also be publicly verifiable.

1.2 Motivation

E2E systems offers radical improvements to security and transparency in elections while maintaining ballot secrecy and accessibility, yet there are few implementations. Even fewer have reached the point in their development to be more than prototypes or curious instructional toys, and none of these implementations are suitable for adoption in regular public and private elections. No E2E verifiable election systems are regularly used in modern elections.

We are motivated to identify and resolve the issues surrounding the lack of suitable E2E implementations. We ask the following questions: Is it possible to make viable, large-scale E2E voting systems for modern elections? What are the key considerations when building a workable E2E system? How will voters react to such a system?

We believe that practical, usable E2E election systems are possible, but they require a high-quality, complete implementation and testing in real-world elections. A practical implementation requires modification of traditional election equipment to meet the assumptions and capabilities of the design. The design must consider and compensate for practical limitations of equipment, election officials, and voters while maintaining security properties.

1.3 Scope

We examine the issues surrounding E2E voting systems in the context of our implementation and field studies of the Scantegrity voting system. We believe that the Scantegrity approach is the most promising E2E proposal to date, because the E2E properties of the system are optional for voters, minimize the work of voters and elections officials, and work with pre-existing election equipment.

Scantegrity is an enhancement for existing optical scan paper ballot systems. It provides E2E properties by allowing voters to note codes on the ballot and check for them on an online public record. Each voter marks her ballot as with a conventional optical scan ballot, but using a special ballot marking pen. Marking a selection with this pen reveals an otherwise invisible pre-printed confirmation number, similar to confirmation numbers already widely used and provided by airlines, hotels, and government offices.

Voters can look up their ballot serial numbers online and verify that their confirmation numbers are posted correctly. Correctness of a final tally with respect to the published confirmation numbers is proven by election officials in a manner that can be verified by any interested party. Thus, compromise of either the chain of custody or the system software cannot undetectably affect the integrity of an election.

1.4 Contributions of this Dissertation

This dissertation contains the findings of over five years of effort in the research on practical verifiable election systems. The contributions of this research include an implementation of an E2E verifiable voting system that has evolved from a series of system proposals, the results of a mock election designed to test the viability of the implementation in a real election, the results of a case study of a modified implementation in a real election and analysis of voter reactions to the system, and a proposal to add a trusted receipt printer component to the implementation. This section outlines the specific contributions of this author (Richard Carback), who is the chief system architect for the Scantegrity system, a founding contributor to the research project, and the main organizer behind the mock election and field study of the Scantegrity system. This section also identifies colleagues, and identifies published work from which parts of the dissertation are taken.

1.4.1 A Free, Open-Source E2E Voting System Implementation

The implementation of Scantegrity (and its predecessor Punchscan) is free and open source software (FOSS) that is available on the project website.[2] Note that a FOSS implementation does not affect the verifiability of the system. This decision was instead based on the desire to make the voting software more transparent and accessible to the public, and to serve as a reference implementation for others. The majority of the software is distributed under version 2 of the GNU General Public License [68], and the remainder under a modified Berkeley Software Distribution License [141]. For a description and discussion of FOSS, see Feller *et al.* [61].

Due to the open-source nature of the implementation, numerous individuals have contributed to it. In addition to new components, the current implementation still makes use of components from the predecessor Punchscan system.

Punchscan, The Predecessor to Scantegrity

David Chaum first proposed a high-level version of Punchscan in 2005 [36], and he led the research team that created the Punchscan system. The research team consisted of the

[2] http://scantegrity.org/svn/

following researchers: Richard Carback, Jeremy Clark, Aleksander Essex, Kevin Fisher, Ben Hosp, Stefan Popoveniuc, and Alan Sherman.

Chaum and these researchers have published numerous papers on different variations and aspects of the Punchscan system [26, 65, 57, 118]. The Punchscan system proposal and implementation are also described in theses by Popoveniuc [115], Carback [23], Essex [56], and Fisher [64]. Carback, Clark, Essex, and Popoveniuc entered an implementation of Punchscan into the intercollegiate voting systems competition (VoComp) and won the $10,000 first place prize among the four finalists. The VoComp submission document [58] best reflects the Punchscan implementation.

There are several components from Punchscan in the Scantegrity implementation. The largest is a modified version of the cryptographic engine (back end), which was written by Popoveniuc. The user interface wrapper for the engine was written by Carback and Bryan Pass, then modified by Popoveniuc. Parts of the website bulletin board were written and subsequently modified by Carback. Code from other programmers, even if it is still in the repository, is not used by the Scantegrity implementation.

Scantegrity

Our team watched Punchscan in practice through an election at the University of Ottawa [57] and other trials [20]. We determined that Punchscan was too different from the mental model that users had for voting and that it may be too difficult to use.

Essex suggested deviating minimally from a traditional optical scan system, and Chaum presented an early version of Scantegrity [37] in which codes were cryptographically associated with each candidate and printed next to each candidate on the ballot. Popoveniuc, Carback, Clark, Essex, and Chaum later published a refined version of this system [117], which is further refined with additional collaborators Alan Sherman and Poorvi Vora [41]. There is no implementation of this version of the system.

In Scantegrity II (**I**nvisible **I**nk), we added additional contributors Ron Rivest, Peter Ryan, and Emily Shen, and we proposed a modified system with hidden codes that are revealed only after candidate selection [38, 40]. Scantegrity II and its predecessor Scantegrity have the following contributions that distinguish them from other systems that provide end-to-end verifiability:

1. Compatibility with optical scan equipment: Scantegrity and Scantegrity II do not require the replacement of optical scan polling place equipment. Both systems interface cleanly with the underlying optical scan system, requiring only a modified ballot and access to the results from the scanners.

2. Familiar ballot-marking procedure: The ballot-marking procedure is very similar to that for a conventional optical scan ballot. Opting into verification of election integrity is up to the individual voter.

Scantegrity II offers the following contributions over Scantegrity:

1. The most visible change is printing technology that can print codes that appear after marking the ballot. While not the original intent, a side effect of this technology is that the ballot marking pen allows marks only in permitted locations. This side effect also permits some improvements to ballot scanning technology.

2. Scantegrity did not hide confirmation numbers, and all confirmation codes were visible on the ballot. This allowed voters to file spurious disputes concerning which codes appear on the website, and required a tedious dispute resolution process to resolve such issues. If voters cannot guess confirmation codes or chit serial numbers,

a dispute regarding the correct recording of confirmation numbers can be resolved in Scantegrity II without the cumbersome physical proof required by Scantegrity.

3. Scantegrity II makes commitments to multiple Scantegrity back-ends and uses a new audit procedure and backend.

   (a) While the Scantegrity audit procedure reveals some information about individual votes, the Scantegrity II audits reveal no additional information if the cryptographic techniques used are secure, and election officials do not collude to violate ballot secrecy.

   (b) In Scantegrity, the probability that a cheating voting system is undetected decreases exponentially with the number of modified votes. In Scantegrity II, this probability is independent of the number of modified votes, but decreases exponentially with the number of back ends audited.

Because there is only one implementation, we refer to it as Scantegrity in the rest of this dissertation. There are, however, two revisions of the implementation, and both are very close to the Scantegrity II proposal in [40], with the primary exception that we used multiple Punchscan back ends instead of the backend proposed in both revisions. The first revision was used for the mock election and the second was used for the municipal election. The finer details of the differences between revisions are explained in Chapter 4.

Carback led the implementation effort with help from Popoveniuc and programmers John Conway and Travis Mayberry. Popoveniuc modified the Punchscan cryptographic engine (backend), while the rest of the team created other software components that had to integrate with the backend. The main components include a scanner, a website bulletin board, an ink printing system, and an election management system.

1.4.2 Scantegrity Mock Election at Takoma Park

We report our experiences, the lessons learned, and the changes made when using our implementation of Scantegrity in a mock election conducted April 11, 2009. A short summary of this work appears in [135] and a longer version in [136]. Our research team did not have expertise in conducting research with human subjects or statistics. To solve this deficiency, we added collaborators Paul S. Herrnson, who has extensive experience examining how voters use voting equipment, and Bimal Sinha, an accomplished applied statistician.

We measured Scantegrity's performance through questionnaires, unobtrusive observations, and independently-administered focus groups. Eighty voters and all six Takoma Park poll workers filled out questionnaires about their experiences with Scantegrity, including questions about how easy the system was to use and administer and how well they understood and accepted the system. The purpose of this exercise was to demonstrate and tune Scantegrity's capability in preparation for the Takoma Park municipal election in November, and the results helped improve the system for the November binding election.

Carback was the main organizer behind this project. With help from Sherman and Vora, he made initial contact, gave several presentations, and finalized an agreement with Takoma Park to conduct the mock election. Sherman, Carback, Mayberry, and Vora were the major contributors to writing the research papers. Sherman, Carback, Herrnson, Mayberry, and Sinha designed the survey instruments used during the study. Sherman, Herrnson, and Mayberry distributed the surveys to voters. Sherman, Carback, Chaum, Clark, Essex, Popoveniuc, Rivest, Shen, and Vora contributed to the final design of the voting system. Carback, Mayberry, Conway, and Popoveniuc worked on the implementation. Chaum handled printing of ballots.

Several implementation changes followed the results of this study. Popoveniuc modified the backend pseudorandom number generator to use three numeric digits instead of two alphanumeric digits. Carback created modifications to the ink printing system. Carback, Mayberry, and Conway modified the scanning software and the website. Carback and Mayberry created a write-in candidate resolver for use during the election, and integrated it into a centralized election management solution.

1.4.3 Scantegrity in the 2009 Municipal Election at Takoma Park

Due to the performance of Scantegrity at the mock election, Takoma Park allowed us to pilot Scantegrity in its November 2009 municipal election. The municipal election marked the first time that anyone could verify that the votes were counted correctly in a secret ballot election for public office without having to be present for the entire proceedings. Again, Carback was the main leader in this effort, although Sherman and Vora took on more significant roles as communication with Takoma Park become more complex and frequent closer to the election. Carback is also the lead author on both of the papers resulting from this study [24, 25].

There are two key contributions from the pilot of Scantegrity in this election:

1. A case study of the Takoma Park election, describing what was done—from the time we were approached by the Takoma Park Board of Elections in February 2008, to the last cryptographic election audit in December 2009—and what was learned. We explain the engineering process of bringing a new cryptographic approach to solve a complex practical problem involving technology, procedures, and laws.

2. A study of voter reactions to the Scantegrity system. Among studies of E2E systems, it is the first to survey such a diverse group of voters in a binding election, the first

to survey election judges in a binding election, and the first to observe election day events that are largely not managed by the system designers. With regard to other election studies, it is among only a handful that study voters in a binding election.

Although the overall experience with Scantegrity was mostly positive in nature, especially the reaction by voters, the case study and survey also capture the consequence of a number of procedural missteps and shortcomings of the system implementation. Because we studied the system, the voters, and election judges in a real election—as opposed to a simulated election—our findings more accurately reflect the true experience and viability of the system.

1.4.4 A Trusted Receipt Printer for E2E Voting

Scantegrity can benefit from trusted hardware components, which would substantially improve the implementation, and could change prevailing opinions in the community about the lack of efficacy that trusted computing brings to voting when compared with E2E. We believe the trusted computing and E2E approaches are complementary. As an example, we propose a trusted receipt printer and several variations of how it might be used in the election environment.

The receipt printer was the most requested feature by Takoma Park voters and election officials, and it is precisely because it must be trusted to operate properly in the polling place that our research team avoided implementing it for the real election. Our main contributions are two designs of the receipt printer using the Trusted Computing Platform,[3] and an analysis of the risks involved with using the two different designs. The first design is an independent unit that produces images of the markable positions on the ballot, and the second is a scanner

[3] See: http://www.trustedcomputinggroup.org/

add-on that decrypts each ballot to produce plain-text confirmation numbers. Both designs preserve the ability for voters to verify if the receipt prints the same confirmation numbers that appear on the ballot.

This work is separate from the work of the main research team, and is instead joint work by Russell Fink, Carback, and Sherman. This work stems from previous work adding trusted computing to direct recording electronic (DRE) equipment [62].

1.5 Outline

Chapter 2 provides background information for this dissertation. This includes a brief history of election verification throughout history, a discussion of general voting system requirements, an examination of current verification systems, an introduction to E2E voting systems, and a review of related work on usability and field studies of voting technologies. Some text in this section is taken from [23].

Chapter 3 is an introduction to the Scantegrity voting system. It provides a detailed explanation on the implementation and how it differs from the Scantegrity proposals. The text in this section is largely from [40].

Chapter 4 covers the mock election, and provides an introduction to Takoma Park and the agreement our research team made with the city. Our research methodology and the results of the mock election are discussed. We also cover the recommended changes to the election and discuss what was actually changed in the implementation. Most of the text in this section is taken from [136].

Chapters 5 and 6 cover the pilot of Scantegrity at the municipal election in Takoma Park. We examine the election from our perspective in a case study, and analyze the reactions from voters who voted in the election. We also provide a list of improvements to make Scantegrity more effective in an election environment. The text is taken from [24] and [25].

Chapter 7 examines the addition of a receipt printer. We provide two configurations in which the receipt printer could be used, provide designs to integrate both into the larger system, and analyze the risks associated with using this technology in an election environment. This section has not yet been submitted for publication.

We conclude in Chapter 8. Appendix A provides additional figures and tables from our survey analysis in Chapter 6.

Chapter 2

Background

It takes significant planning on the part of officials to make sure that elections are conducted properly. Fraud, failed technology, and unreliable election results have been noted throughout history. Election systems have always been flawed and hard to verify.

In this chapter, we provide a background into the conduct of elections and their verifiability. We include a history of the susceptibility of elections to manipulation in Section 2.1, and an overview of the problems found in modern election systems in Section 2.2. We discuss the requirements and security properties for elections in Section 2.3, and provide an overview of current studies for verifiable elections systems in Section 2.5. We end with a discussion of the predecessor system to Scantegrity, known as Punchscan in Section 2.6.

2.1 A Brief History of Election Verification

It is likely that elections, *i.e.,*events in which a group of people select a person or persons, have always existed in some form, and historical accounts date back to ancient Greece. It is important to understand the historical context in which we are working so that we can understand the traditional expectations of voting systems, their properties, and how secure they were. In this section we discuss different election methods throughout history, explain how these election methods were verified, and explain how they could be manipulated.

2.1.1 Ancient Elections

Some of the earliest examples of elections that we know about happened in the city-states of Sparta and Athens [139, 21]. Votes in these elections were limited to men who had completed military obligations.

In Sparta, an assembly known as the *Apella*, comprised men over 30 years old, elected some leaders by acclamation (shouting) [66], which is considered a form of range voting.[1] Evaluators, in a separate building where they could not witness the election, assessed the loudness of the crowd when shouting for each candidate in the election, and the candidate with the loudest crowd was designated the winner. The order in which the candidates were presented to the crowd was decided by lot, so it was not known to the evaluators.

Verification of this election was by virtue of its transparency—every "voter" witnessed the strength of the shouts from the crowd for each candidate. When the election was close, however, it would have been simple to manipulate the result if an adversary could signal the evaluators in some way (e.g., by clapping in a particular rhythm). Thus, it is not as foolproof as it appears.

Athens is where the word democracy, or *demos*, originates. The city was a direct democracy, and all male citizens completing military service were permitted to vote in an assembly called the *ecclesia*. As such, they did not hold regular elections, but they did have elections to grant citizenship, select leaders for important positions, and select prominent

[1] For a discussion of ancient Spartans and the particular advantages and disadvantages of their form of range voting, see the website by Warren Smith: `http://rangevoting.org/SpartaBury.html`.

citizens for 10-year-long ostracisms. Most voting was done by raised hands, but close votes would be difficult to count with over 6,000 members in the *ecclesia*. One alternate voting method had each citizen put a black or white stone representing each side into an urn.[2] The side with the most stones won. In ostracism, voters wrote the name of the person they wished to ostracize on ballots made of broken clay jars; this negative vote is the same as writing a name on a piece of paper [21].

Both methods are reasonably secure against manipulation, and the verification properties are similar to hand-counted ballots in a room: voters have to watch what is happening. However, there would be a large unknown number of voters (in both cases, the quorum was 6,000 members), making it difficult to keep track of the voting. Thus, it would have been fairly straightforward to manipulate the election by stuffing the ballot urns or adding additional ballots during the tally.

2.1.2 Early Modern Democratic Elections

Election laws in early democracies were not well defined, and were not consistent between election jurisdictions. Voting rights were generally limited to men, with numerous additional restrictions (e.g., land ownership, age, freemanship) [28, 90, 77, 15]. Paper was available, so elections were generally conducted by a clerk who wrote down the name and vote of each eligible voter during a voting period, or by an assembly during which a show of hands was counted.

In theory, these elections are secure against manipulation. Anyone who wishes could find each voter on the roll and ask him for whom they had voted. In practice this could be quite difficult, depending on the locality. Some municipalities—e.g.,western frontiers in

[2] The term *blackballed* originates here, as some modern clubs often use a similar method to accept or reject new members. One (sometimes more) black stone causes the member to be rejected.

the United States—could be quite large, and in others, land ownership (but not necessarily residency) might entitle someone to vote. In large municipalities, verifying the election results meant finding thousands of people. It could be quite costly, and very few candidates would have the resources to carry out an effective audit of the election.

Since voter choices were public, voter coercion was also a serious problem during elections. In England, as late as the 19th century, there were "pocket boroughs" (municipalities that elect one or more representatives), which provide an extreme example. These pocket boroughs were completely controlled by wealthy individuals, and could even be bought and sold [28, 90]. This control was possible because all votes were public and the boroughs were small. Challenging the result of an election was unlikely to change the result, as it was easier to bribe and/or threaten the small number of voters in these boroughs.

When power and wealth are equally distributed among the voting members, coercion might not have been an issue, but coercive environments were prevalent everywhere and appear to have been normal. Voting rolls could be used to make lists and punish people. Candidates and their supporters used bribes, alcohol, threats, family and anything else that could be imagined to persuade voters, and, good or bad, often followed through with their promises. In times of strife (e.g., during the U.S. Civil War), not voting for the ruling party was often considered treasonous [15]. Having the votes be public proved to be a problem for selecting representatives who represented the people as opposed to those who represented the most powerful members of society.

2.1.3 Introduction of the Secret and Australian Ballots

The secret ballot was first codified in France in 1795 [77]. In this style of election each voter wrote the name of the candidate he wished to vote for on his own ballot paper. He would then take the ballot with him to the polling place, sign in with the election clerk, and cast his ballot in a sealed ballot box. This required the voter to be able to write or to find someone who could produce the ballot for him.

The Australian ballot was introduced in the Australian colonies in 1856 [104]. Australian ballots are intended to be secret ballots that are printed by the government and distributed and marked at the polling site. In the original system, each voter crossed out the candidates for which he did not want to vote, as opposed to the modern practice of making a mark next to the candidate to be selected.

Australian ballots offer several advantages over other kinds of secret ballots. Primarily they reduce the amount of work for voters, because the voter need only recognize the name of the person for whom he wishes to vote, and need not know how to write. Because the printing is the same on every ballot, it simplifies the counting of ballots by the election administrators and makes it more difficult to generate fake ballots.

Assuming the ballots are protected properly, secret ballot elections offer verification advantages over non-secret elections. Auditors need only consult the paper record instead of contacting each individual voter. This process is much less costly.

Unfortunately, it is relatively easy to manipulate election results in secret ballot elections. Attackers need only to be able to create ballots and create fake seals. The number of voters who vote can be checked against the totals, but there is very little that can be done to protect ballots from being replaced after they are deposited into the ballot boxes other than very vigorous oversight by the opposing sides in the election. Such oversight is not always available.

The implementation of secret ballot election systems is often not very secret, because information that can identify voters can be added to the ballot. One example is England's ballot act of 1872 [51], which introduced the Australian ballot in that country. It requires a serial number on the back of the ballot and counterfoil stubs that can be used to identify voters. This counterfoil process is still used [17]. Also, any time that voters can write on the ballot they can uniquely identify themselves; even if they are not allowed to make unique marks, it is still possible to make unique marking patterns that cannot be recognized as unique marks [121].

2.2 Verification in Modern Day Elections

Many modern-day elections still use hand-counted Australian style paper ballots, but since the turn of the 20th century there has been a constant drive to push for automation that can increase the security and accessibility of elections. Universal suffrage is now the norm, so any adult citizen can vote. The number of voters and the things for which they vote has increased to the point where managing counting processes and polling is very difficult, so election officials constantly look to be more efficient through automation. Now there is also a significant need to accommodate voters with disabilities and give them the same ability to vote privately with the same security features as other voters.

The past century has seen numerous types of voting machines, such as lever machines and punchcards [131, 54]. The general approaches are to have a machine record the votes directly or have a machine read in a physical record marked by voters. There are also some countries that use the Internet to receive and record ballot selections [94]. Each of these approaches has serious obstacles to its integrity and privacy, and none provides much ability to verify that a result is correct. In the next sections, we discuss the more recent and popular implementations of each approach—DRE machines and optical scanners—and explain attempts to verify that these systems are operating properly.

2.2.1 Direct Recording Electronic (DRE) Equipment

DRE systems, also called electronic voting machines (EVMs), are independent voting computers that directly record intent by the voter and report their results. They offer advantages in accessibility, from touch screen to audio to puff and sip style interfaces. The security of the ballot is the same for all voters who use the system.

Current DREs have consistently been shown not to support even basic protection mechanisms on their records and software [91, 30, 133, 70, 19]. This issue is problematic, as they usually store aggregate counts rather than full ballots for each voter. Thus, some counting is done on the machine, and totals from each machine are added with those from the other machines.

Even in their simplest form, DREs have been shown to be susceptible to modification through replacement of the software or replacing the key components that allow them to report their results. A recent study of Indian EVMs [144] shows that, among other things, the simple display could be easily replaced to report whatever results were wanted by an attacker. Thus, even if it is possible to secure the software, the hardware must also be secured from attackers.

There is very little verification that can be done using a DRE by itself. An undetected corrupt DRE system could have a large impact on the election results, and an attacker who could corrupt one of the systems could likely easily corrupt them all. Various techniques and approaches exist that intend to detect malicious machines such as parallel testing during election day, voter-verified paper audit trails (VVPAT), independent verifier machines, and software review.

Parallel Testing.

Parallel testing is a technique that is intended to uncover a malicious attack on the election system during the voting period. It is specifically designed to detect a machine that would otherwise recognize when it was being tested [29].

To conduct a parallel test, officials unpredictably (randomly) select a subset of the machines that will be used during election day, and use each machine in a simulated election environment during the election day voting period. The "test voters" use the same election day configuration as in the real election, and the results of the parallel test election are known to the testers. The frequency with which the "test voters" use the machine is also simulated so that times of the real election where voter activity is heavy (e.g., in the morning, after lunch, and dinner time) are the same.

Parallel testing is often touted as the "...only procedure available to detect non-routing code bugs or malicious code on DRE systems" [67]. For being so highly recommended, it is incredibly trivial to bypass, and may not be very effective unless a large number of machines are corrupted and a large number of machines are tested [7]. In the past, parallel testing has not been implemented in its "pure form" [108, pg. 101].

Parallel testing is security theater for various reasons:

1. **It is trivial to bypass.** If an attacker can find out which machines will be tested, he can bypass the test. If he cannot, he can design a trigger that will permit him to activate the machines that are not being tested on election day [87]. While this attack requires a wireless receiver or voters who are willing to attack the system, it is reasonable to assume that an attacker capable of corrupting the DRE machine would also have these capabilities. It is also feasible to assume that an attacker might do the opposite, and insert some pattern recognition in the DRE software to indicate that

it is being tested or modify what happens upstream from the software (single-digit changes in voting totals could go unnoticed by observers).

2. **It is difficult to simulate real elections.** There are no studies on how one might simulate machine usage such that it is indistinguishable from usage during any given election. Because they are familiar with the machines, testers would be prone to cast votes much more quickly than regular voters or not make the mistakes that voters tend to make. A determined adversary could determine these differences and make conservative guesses as to whether a test might be conducted on the machine. The adversary could also influence the voters themselves to change behaviors on election day that are not known to the testers (e.g.,to use an unknown write-in candidate that a tester would not know to use).

3. **It creates additional vulnerabilities.** An attacker who knows that the election will not go his way could corrupt the parallel testing process to show problems when there are none. This attack would throw the election results into doubt.

4. **It does not audit machines used in the election.** Thus, it requires that additional machines be purchased solely for the purpose of testing. It also adds additional costs to acquire and train the testers for election day. Since the tests might find nothing, it can create a feeling that the test is pointless and discourage future conduct of the test [87], especially when the tests are passed, but discrepancies (due to ballot formatting or other issues) appear during the election.

5. **It does not protect against insider threats.** An insider could make any of the attacks described thus far much more effective. He could control who carries out the tests, what memory cards they use, what procedures they use to conduct the test, and the

selection process. Any of these could be used to tell the machine that it is being tested and should act properly.

6. **There is no recovery mechanism.** Parallel testing is only a detection mechanism. If a problem is detected, additional malicious machines cannot be detected. There are no procedures for what happens when a problem is found.

Even if we assume the process is perfect, parallel testing does not live up to its purpose. It cannot guarantee that malicious code on a DRE system will be detected. At best, it merely makes attacks against the system more costly to an attacker.

Voter Verified Paper Audit Trails (VVPATs)

A VVPAT is a paper record that is printed before the voter is permitted to cast the ballot. The voter is asked to cross-check the printed record with the record on the DRE and confirm that the two records match her intent [95]. The act of accepting the printed record is what makes this method "verified" as opposed to "verifiable." The VVPAT method is a *software independent* method as described in section 2.3.3.

VVPAT systems suffer from a fatal flaw: a voter cannot prove that his vote was changed or that the machine is acting maliciously. The DRE need only ignore the voter's selections, show whatever it chooses on the final review screen, and print a matching receipt. If the voter detects the error, the machine can simply act properly when the voter goes to correct the errors. This behavior would likely lead the voter to believe that she had made the initial mistake, and she might not complain that the machine did not act correctly. On the converse side, voters may make legitimate mistakes and complain anyway, so it is indeterminate, from the election official's point of view, who should be trusted.

In practice, this attack is devastating. Tests indicate that fewer than 40% of voters will check a printed record and discover a problem [60]. This holds true when entire contests are removed or added to the ballots! Additionally, voters who do not notice problems can be characterized by their usage of the machine, making the attack even more effective by targeting only those voters who are unlikely to notice a discrepancy.

There are other problems with VVPAT:

1. **A DRE must be able to "invalidate" a ballot if the voter changes her mind.** Thus, the DRE has additional choices. It could permit the voter to make the changes and then, instead of accepting the ballot, reject it, print what it wants, and accept that ballot. Alternatively, it could simply wait for the voter to leave, reject the ballot, and make up an acceptable vote. This attack can be solved through protocol, but it is unlikely that a voter would know what the protocol is while voting. Even when the voter detects the issue, she again has no proof, so election officials may not believe her.

2. **The printer may malfunction.** There are many moving parts, making this component more likely to fail than the DRE. The printer could run out of ink or paper, and there might not be supplies at the polling site. Election judges might not be able to solve the issues. A printer failure adds an additional failure mode to the DRE and increases cost.

3. **Auditing may not be sufficient.** Someone will have to audit the paper trails of the DRE to make sure they match the reported results. This audit can be manipulated or broken in the same ways the parallel testing audits can be subverted.

4. **It creates additional vulnerabilities.** An attacker who knows that the election will not go his way could simply change the results to mismatch on purpose. This attack

would again throw the election results into doubt. Also, most implementations of VVPAT use receipt paper tape rolls, preserving the voting order and posing a risk to voter privacy.

5. **There is no recovery mechanism.** Just like parallel testing, VVPAT is largely a detection mechanism. There is no set process for determining if the machine or the hand count are ultimately correct; however, VVPAT is better than parallel testing because it produces a separate record and it can localize the problem to a specific voting location or machine.

Redundant records are not always a good thing. In general, having multiple records may make an attacker's job harder, but the attacker only has to change the record that will ultimately be used and/or trusted (not necessarily both). Also, redundancy can work against a system, as changing a digital record in an obviously malicious way may allow time for a more subtle manipulation of the physical record. Since a VVPAT is a separate record, it increases the area of attack on the system as a whole.

Independent Verifiers.

Similar to the VVPAT, an independent verifier (IV) creates a separate record from information provided by the DRE, except that it is an independent device created by a separate vendor with a different voter interface. Unlike VVPATs, an IV can produce a digital record for voters with disabilities. Thus, a voter who cannot see could listen to an audio record to verify her vote.

An IV would be one simple way to add an E2E verifiable audit to a DRE election system; however, IVs suffer from the same problems found in VVPATs and parallel testing. The equipment can malfunction. Separate records create additional vulnerabilities. There are no recovery mechanisms, and there is no way to determine which record should ultimately be trusted. Most importantly, voters may not accurately compare the two records.

IVs are vaporware. They are not being used, and there are no products that are fully developed and integrated with current DRE systems [137]. In terms of security, the specifics of the IV implementation can affect different aspects of the integrity and privacy of the system, and would require different trust. Because an IV may be entirely software based, it may be slightly worse than a VVPAT because it can be corrupted, and again, it is not clear which redundant record should be trusted when a discrepancy occurs.

Software Review and Software Verification.

A traditional approach to detecting a corrupt machine would be to permit review and verification of the source code of the software. Unfortunately, software in current implementations is considered proprietary, and there are insufficient mechanisms in current implementations to verify that the correct software is running. Even if the correct software can be verified, there may be a malicious element that was not caught during the review or a vulnerability that makes it possible to corrupt the machine or the results it produces.

Unlike the other approaches we have discussed, this approach has promise if the problems associated with it can be solved. There are two orthogonal approaches that have been considered but not implemented: using a Trusted Platform Module (TPM) [62], or creating a DRE with E2E verifiable properties [34, 101].

A TPM can permit users to verify that the correct software is running, and voters could do this before using the machine. There are, however, limitations to this approach. There may be vulnerabilities in the software that could be exploited or hardware modifications that could still change the vote. The voter also cannot know if her vote is included in the final tally, although there could be some evidence as signed records for each machine could be publicly posted.

An E2E DRE system would make it possible to verify that each vote is properly included in the tally, and could achieve properties that are as good or better than Scantegrity. However, to date, such proposals are very complex and may require a particular human interface protocol where certain actions must happen before others to maintain the integrity in the system. It is doubtful that voters would detect subtle changes that might permit the wrong vote to be recorded [86]. And again—as we have seen with all the other methods—it may not be clear if the voter or the machine is the problem when a voter complains.

Assuming these problems can be overcome, and E2E or TPM DRE (or both) could have superior usability and accessibility compared to a paper-based E2E voting system like Scantegrity. It could produce receipts digitally and prevent mistakes like over- and undervotes. Unfortunately, DREs are generally more costly up front than paper systems, and the additional complexity involved in creating these advanced DRE systems may make them a poor choice compared to paper, even considering the additional costs involved in making paper accessible to voters with disabilities.

In the United States, most states and counties are moving toward optical scan systems [113]. This trend is one of the reasons we decided to use an optical scanner as a base for an E2E system. The other reasons include making something that preserved what voters already had (a paper record) and supporting the same mental model to which voters have already been exposed in optical scan systems. An E2E DRE system and the concepts behind it appear too different to easily gain acceptance from voters.

2.2.2 Optical Scanners

In an optical scan system, the voter fills out a paper ballot and enters it into the machine. Optical scanners can be used to count ballots in a central location, known as central count optical scan (CCOS), or—in the more popular form—in each precinct, known as precinct count optical scan (PCOS).

The advantages of an optical scan system are that voters do not have to learn how to use a machine or a DRE system, it is robust when the equipment fails, and it cannot modify the physical record. Optical scan systems are *software independent* as described in section 2.3.3, because they do not create the record like DREs do: they merely report it. Thus, an optical scan system is inherently a more secure election system than a DRE [85].

If a scanner fails, the ballots run through that scanner can be sent through another scanner, which is why the system is robust. Most scanners do not have hardware that can print on the ballot, so it is not possible for a scanner to modify the selections on a ballot. Despite these advantages, however, optical scanners are still prone to fraud and ballot stuffing and they do not add substantial verifiability over traditional paper ballots.

The other disadvantages of an optical scan system are its poor usability and accessibility. Voters who cannot see the ballot need an interface in which they can mark the ballot accurately, and this interface may have its own security problems. Voters with other disabilities may have trouble marking the ballot and may not be able to read it or send it through the scanner. If voters make mistakes or the marks are unclear, they will remain so in the ballot box, and any recount will have to handle these discrepancies.

Security and Privacy with Paper Ballots.

Most of the security problems in optical scan systems are orthogonal to the scanner because the security is derived from the security of the paper ballots. The assumption is that the paper record will not be modified and that the intent of voters can be determined. Therefore, the ballots can be referenced to resolve a discrepancy [83]. This assumption is naive, and the fact that a voter creates the record can cause additional issues:

- **Ballots could be replaced.** An adversary with access to the ballot store could replace the ballots with his own to influence the result.

- **There is generally no way to tell the difference between ballot marks from a voter and an adversary.** If the adversary gains access to the ballot store, he could vote in contests which the voter chose not to mark, or he could selectively overvote in contests where his opponent is chosen, invalidating the vote.

- **Voters can be coerced to chain vote.** An adversary who gains access to a ballot before the election can fill it out, tell a voter to cast it, and return a new blank ballot to the adversary.

- **Voters or the ballots can contain additional information to violate voter privacy.** An adversary can force a voter to mark a ballot in a particular pattern or use a unique marking style to identify herself, or there may already be existing information on the ballot that the voter can provide to a coercer to identify her ballot. The adversary can then look at the paper ballots to verify the voter has complied with his demands.

- **Rules for counting unclear marks and ballot styles can bias election results.** They also may not be applied consistently when performing a hand count.

When using the optical scanner in addition to paper ballots, it can produce fast results but can also introduce additional problems. The optical scanner acts as a redundant record for the paper, so there is the question of which record should be trusted (although in the optical scan case, it is usually the paper). Because the paper is typically not counted, only used as an audit, the attack space is much larger in an optical scan system, as an adversary can choose to attack either the audit or the optical scan system. A discrepancy in either system gives the adversary more time to modify the paper and change the election result.

In an optical scan system, the voter has no way to verify that the ballot she cast is being counted properly by the optical scan system. It may simply ignore it, or record—maliciously or unintentionally—a result that is different from the one she intended. In practice, simply modifying a configuration file can change how the ballots are recorded and these are typically not well protected [84, 89]. The only way to detect this sort of malfeasance is through an audit, which may not be conducted at a sufficient level to detect a problem.

Unique Identifiers, Digital Signatures, and Other Approaches to Improve Security

Various techniques can be used to enhance ballot security in an optical scan system. We considered several while designing Scantegrity.

Placing unique identifiers, or serial numbers, on a ballot can identify when ballots have been replaced by an adversary and help determine why a discrepancy has occurred between the optical scan counts and audit. Unfortunately, serial numbers pose a threat to secrecy, but a serial number does not affect the threat model. If we assume that a voter will cooperate with an attacker to reveal ballot selections, there is no way we can prevent the voter from adding identifiable information to the ballot when voting. Also, serial numbers can be added in a form that would be difficult for a voter to report back to an adversary (e.g., a barcode format that is difficult to reproduce or read).

Digital signatures can be printed on the ballot that provide the same guarantees as unique identifiers, with the additional guarantee of verifying the state of a ballot after it has passed through the scanner. This prevents modification of a ballot after the election. It also does not suffer from the privacy problem that a serial number introduces. Unfortunately, a corrupt machine could add marks to the ballot that, while detectable, might not be noticed.

Another set of approaches concern how the ballot is scanned. If the scanner is simply an image sensor that is broadcast to multiple independent parties, it may be possible to get better integrity. Such approaches are limited to scanning immediately after the election for privacy reasons, and have been proposed with video and still shot cameras [142, 2]. These approaches can be very effective, but still do not provide a voter any way to verify that her vote was scanned at the end of the day. Broadcasting the ballot images in such a public way will also increase the effectiveness of various voter coercion schemes in which voters are instructed to vote in patterns or to select unique write-in candidates.

While there are other E2E approaches that use optical scan equipment, to our knowledge Scantegrity is the only such approach that preserves the ability to hand count the ballots after the election. Because this hand count capability and other features of optical scan elections are preserved, Scantegrity inherits many of the advantages and disadvantages typically associated with optical scan systems. Scantegrity solves or improves some of the problems associated with optical scan, but not all.

2.3 Election System Requirements and Properties

A comprehensive set of requirements and properties would be an unwieldy topic for this section.[3] Instead, we will limit our discussion to important high-level requirements and

[3] See, for example, the several-thousand-page Voluntary Voting System Guidelines [140] document published periodically by the U.S. Election Assistance Commission.

properties and what we considered in the design of the Scantegrity voting system. Many of these requirements and properties were also important in earlier system designs [23].

Our high-level requirements when designing Scantegrity were that it should support E2E verifiability, that it should maintain the secret ballot, and that the voter experience should be as close to optical scan as possible. Each of these requirements met a need we felt was important. The integrity of the cast ballots should be protected, and current systems do not provide anything like E2E verifiability. The other two goals are to preserve the attributes of an optical scan voting system and ensure that voters who do not know or care about verifiability can still use the system.

There are also a number of other requirements common to voting systems that influenced our decision. In the next section, we discuss these general criteria, and in the following sections we discuss the properties that we focused on achieving with the Scantegrity voting system.

2.3.1 General Requirements for Voting Systems

Public systems are costly and meant to last for as long as possible. If a failure occurs, there is relatively ample time and opportunity to fix the problem or replace the system. Voting system requirements are unique because a voting system is a large-scale system intended for use by the public for a short time that has little tolerance for failure; otherwise, public confidence in the process is severely damaged. An election is rerun only if there is catastrophic failure and a serious threat of open revolt. Election systems require a higher level of assurance than other systems.

With this in mind, good election systems meet the following general requirements:

1. **Integrity.** It should not be possible to modify, replace, add, or remove ballots. Ballots should be cast as they were marked, recorded as cast, and tallied as recorded.

2. **Privacy.** The system should support a secret ballot, and each voter should be able to make her selections in private without fear of being identified.

3. **Transparency.** Generally, each voter should be able to understand the basic operation of the system. Each part of the system must also be observable and verifiable.

4. **Usability and Accessibility.** The usability and accessibility requirements for a voting system are the most extreme of any system. Voters are not required to read a manual or be familiar with an interface, and they may have any number of disabilities. Instructions at the polling place must be minimal, each voter must be able to mark her ballot as she intends, the system must be intuitive, and it should cater to voters with vision, hearing, and dexterity problems.

5. **Ease of Administration.** Most of the work done in elections is accomplished by volunteer election judges who are paid a token wage. Often, the only requirement for an election judge is that he or she be able to read and write in the native language. Like accessibility, this requirement puts the threshold for ease of managing the system at an extreme level relative to other systems where professionals can be paid to operate and maintain the system. Administering the system should be obvious enough that election judges are able to set up, operate, and close the polls with minimal technical knowledge.

6. **Cost and Durability.** Voting systems are typically used over the course of several days every other year, and it is hard to justify an expensive system. The system should be as cheap as possible, and the equipment should be built to last as long as possible.

7. **Scalability.** The system should scale in several different ways. The number of voters supported should grow as more equipment is added, and the tally should still finish in a reasonable amount of time after the election is complete. This timeliness requirement should also be true for any number of candidates per race and number of races per ballot. The system should also not have technical limitations preventing it from implementing election rules.

8. **Reliability.** It must be reliable and resistant to disruption, e.g.,failure of the power grid. Any voting system must support a way to continue operation without power. Barring natural disaster or violent military actions that keep voters away from the polls, the system must be available.

9. **Resiliency.** There should be ways to recover from catastrophic failures. In many cases, this is a matter of redundancy and procedures (e.g., publishing the current count of votes for each time period). The difficulty here is that you wish to achieve recoverability without drastically affecting the privacy, the rest of the system, or putting a lot of responsibility on volunteers.

Optical scan election systems meet most of these criteria reasonably well. They are obviously scalable, reliable, and recoverable because they are paper-based. You can print more ballots if there are more voters, ballots can be scanned later if a scanner fails, and the paper or digital records can be consulted if something goes wrong.

The paper incurs higher recurring cost, but the precincts do not need as many scanners as they would DREs. Optical scanners are also simple machines that are durable for multiple years of use.

Operation of the scanner should be relatively straightforward for the average person, so scanners are transparent and easy to administer. Storing paper ballots securely may be an administrative hassle, but better trained salaried officials can be responsible for this task. Likewise, marking a ballot is straightforward. Voters do not need to learn to use a DRE or other special interface to use the system, and election officials can provide paper examples of how to properly mark the ballot.

Scanners do not fully meet these requirements in a number of ways. As already discussed, the integrity is poor and they are not transparent after election day. An observer cannot always watch what happens to the ballots after the election, and the way the software operates on the scanner may not be known. Well-protected paper records can provide some protection, but outside of being able to verify that a ballot was cast as marked, there is very little guarantee that it was recorded as cast or tallied as recorded. Paper records do have the property of software independence; however, to satisfy this requirement, we need systems to have E2E verifiability properties that provide verifiable integrity and give transparency of every step in the election.

An issue that stems from the use of paper is privacy. In theory, this protection should be absolute. There should be no way to violate voter privacy. In practice, however, many "secret ballot" systems preserve the ability to determine voter choices (e.g.,[17]), and there are no paper based (or optical scan) systems that provide privacy when the voter cooperates with a coercer. It may not be possible to offer complete privacy protection in a voting system [79], but voting systems should at least be coercion resistant, and any materials given to the voter should not permit the voter to prove how she voted.

2.3.2 E2E Election Verifiability

E2E verifiable election systems were first mentioned in the 2005 Voluntary Voting System Guidelines from the U.S. Election Assistance Commission [140], and later clarified by Popoveniuc *et al.* [120]. E2E is understood to be integrity-centric and does not consider voter privacy or other properties, although these other properties are important in election systems.

As specified by Popoveniuc *et al.* [120], an election system is considered to be E2E verifiable when we can verify the following six properties:

1. **Presented ballots are well formed.** At any time after the election, the voter can detect if a vote about to be cast does not represent the selection for the candidate(s) she intends with some nonzero probability. In the optical scan context, this means that she can verify that the information printed on the ballot matches how the optical scanner will interpret the ballot, by, for example, spoiling the ballot she was going to use and being permitted to take it home for later inspection.

2. **Cast ballots are well formed.** If a cast ballot will not be counted (due to over or undervotes, or negative votes in some systems), anyone can detect that it was incorrectly formed with high probability.

3. **Ballots are recorded as cast.** At any time after the election, the voter can detect if her ballot was recorded improperly with some nonzero probability.

4. **Cast ballots are tallied as recorded.** At any time after the election, anyone can check, with high probability, that the tally is the correct computation from the recorded ballots.

5. **Tallied ballots are the same set as those that were recorded (*i.e.,* consistency).** At any time after the election, anyone can check with high probability that the recorded ballots are the same as the tallied ballots.

6. **Each ballot is subject to a "recorded as cast" check.** At any time after the election, anyone can check that each recorded ballot belongs to a unique corresponding voter.

It is required, for each of these properties, that the system provide an irrefutable public proof of malfeasance for the person who performs each check. Also, whenever there is a protocol that must be followed to ensure integrity or perform these checks, there must be public proof if the system does not follow the protocol. For example, if a voter wishes to audit a ballot, the system cannot ignore the request and cast it instead without providing proof to the voter.

2.3.3 Software Independence

Rivest and Wack propose a general property for acceptability of voting systems known as software independence, which is related to (but is not the same thing as) E2E verifiability [123, 125]. Simply stated, an election system is software independent if *an undetected change or error in its software cannot cause an undetectable change or error in an election outcome.*

Depending on the threat model, a system that is considered to be E2E may not necessarily be SI. Consider Helios, which is considered E2E [120]. When the voting workstation is compromised, the system may be able undetectably to change the choices made by the voter [59].

To date, a software independent system is typically understood to use a paper record that the voter can verify before casting her ballot. Thus, for example, a DRE with VVPAT, a modified version of Helios to print the ballot encryptions before casting, or an optical scan system would meet this property.

2.3.4 Coercion Resistance and Receipt-Freeness

Coercion resistance, otherwise known as voter privacy, is the inability of someone other than the voter to know how she voted. This property must be inherently limited because, ultimately, the voter must transfer her intention to voting officials, and it is not possible to guarantee that she cannot be watched during this transfer. Election officials could set up monitoring that the voter cannot detect, or a coercer could force the voter to use recording technology that the election officials could not detect.

Future technologies may make this situation even bleaker, and it is important to take into account the capabilities of a voter or attacker. If we assume that a voter could have some sort of undetectable futuristic ocular implant that allows an outsider to see what she saw, how could we reasonably protect against such a thing without assuming a similarly futuristic device is available to combat this threat?

Thus, it is debatable how strong coercion resistance should be for a voting system so that it is a useful, attainable goal. At a minimum, voters should expect to be reasonably protected from third party thugs that are not involved with the voting process, which appears to be what current systems try to accomplish. After that, the degree to which an insider must be involved to violate a voter's privacy and where the threshold should be set is unclear because so much of a voting system's privacy depends on the environment in which it is used.

A reasonable goal for a voting system to meet is that it be receipt-free as defined by Benaloh *et al.* [14]. Receipt-free means that the voting system should not give the voter something that would allow the voter to prove how she voted. Receipt-freeness does not prevent the voter from providing information that reveals her ballot, but it makes a coercer's job more difficult. For further discussion on incoercibility, see [18, 96, 79].

2.4 Different Approaches to End-to-End Voting Systems

E2E voting has its origins from over two decades of cryptographic research. The use of cryptography in voting originates in 1981 with Chaum [32]. The ensuing decades saw the introduction of many electronic voting systems using cryptography to improve security and privacy guarantees (for a nice survey, see Adida [3]). These original schemes focused on the voting over the internet with trusted desktop machines, and only more recently have E2E schemes emerged where voters use paper ballots and/or obtain paper receipts from a polling site.

The first group of proposals that could be called E2E include protocols by Chaum [35] and Neff [101], which were partially implemented soon after (Chaum's as *Citizen-Verified Voting* [44, 78] and Neff's by VoteHere). Several more proposals with partial implementations or prototypes followed: *Prêt à Voter* [43], *Punchscan* [65, 118], the proposal of Kutylowski and Zagórski [92] as *Voting Ducks*, and Simple Verifiable Voting [11] as *Helios* [4, 5], *Civitas* [99], and *VoteBox* [132]. There are also several other proposals that claim all or some end-to-end properties, such as ThreeBallot [126, 124], and *Scratch & Vote* [3, 6],*ClearVote* [116] *Bingo Voting* [18] and *Split-Ballot Voting* [98].

This section provides a brief overview of the different approaches used to support E2E verifiability while preserving receipt-freeness. In each case, the systems create an anonymized structure or protocol that can hide the input votes from the final tally but still permit verifying the election. Most systems also include the notion of an electronic "bulletin board" where data are made public in an irrefutable and unalterable manner (see [114] for a discussion of bulletin boards). The three general techniques for creating E2E voting systems are non-cryptographic protocols, homomorphic encryption systems, and mix networks. We describe each in the following sections.

2.4.1 Non-Cryptographic Protocols

There exist four proposals using this method: Farnel, ThreeBallot, Vote-AntiVote-Vote (VAV), and Twin. Each proposal allows the voter to hide in the noise of the system while still providing a checkable receipt.

Farnel [8, 52] and Twin [126] are based on the concept of "floating receipts." The voter can vote as normal with a paper ballot that contains a serial number. After voting, the voter casts her ballot and receives a copy of a randomly selected previously cast ballot. If all voters verify their receipt ballots and we can trust the random selection process, there is a high probability that modification of any ballot will be detected. Farnel is slightly more complex than Twin, as it does not have take-home receipts. Instead, a voter would sign the received ballot and cast it into a second ballot box.

ThreeBallot and VAV [126, 124] use multiple-cast ballots to allow voters to hide in the noise. In both systems, the voter fills out each of the multiple ballots, a machine is trusted to check that they have been filled out correctly, and the voter takes one of the ballots home with him as a receipt.

In ThreeBallot, the voter receives three identical ballots. To vote for a specific candidate, she marks exactly two of the mark positions for the candidate she selects on her ballots. For the other candidates that she does not choose, she selects exactly one of the mark positions on any of the ballots. All three ballots are cast and a copy of one of the ballots is chosen by the voter.

In VAV, the voter receives two ballots and one "anti-vote" ballot. One of the ballots must match the anti-vote ballot, and the voter may take either of the regular ballots as a receipt.

2.4.2 Homomorphic

Homomorphic encryption is a cryptographic system that permits the ciphertexts to be added together to produce a summation that, when decrypted, produces a sum of the plaintexts. Suppose we have an encryption function E, decryption function D, key k, and numerical plaintexts p_1, p_2, then:

$$D_k(E_k(p_1 + p_2)) = D_k(E_k(p_1) + E_k(p_2)) \tag{2.1}$$

For our purposes, it is not necessary that the ciphertext be the same for both encryptions, only that the decryptions always produce the same result. Also, homomorphic systems are typically public-key systems [100], which provide an advantage as generating the ciphertext can be done by a third party without giving them information that might decrypt other ciphertexts. Examples of cryptosystems with this property are Paillier [109] and ElGamal [55].

2.4.3 Mix Network

Introduced by Chaum [32, 33], a mixnet decrypts a set of messages such that no one individual is able to determine the sources of the messages. This is analogous to voting in which paper slips are put into a box and it is shaken, effectively mixing the ballots so that people cannot trace ballot papers back to voters. Such mixing systems are more useful if they are robust or verifiable [102, 34], meaning they can permit an observer to verify that messages were not added, deleted, or modified by the system.

A mixnet comprises a series of nodes that partially decrypts input messages and apply a random permutation to the order of the messages it receives before sending them to the next node in the mixnet. To create the messages, the sender uses the public key, p, of each node

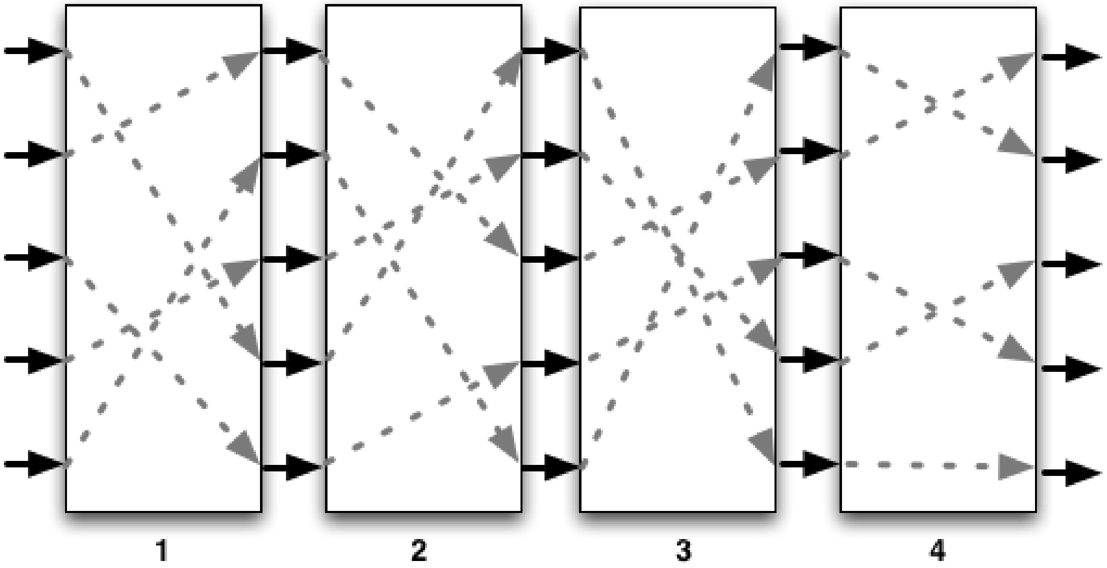


Figure 2.1: **Mixnet**. Each of the four nodes in this example chaumian mixnet partially decrypt input messages, applies a random permutation to the order of the messages, and sends them to the next node in the mixnet. Messages are created for input to the mixnet by successively encrypting the message with the public key of each node in reverse order to hide the source of each message from an observer.

and successively encrypts the message in reverse order. For example, with a mixnet of four nodes as in Figure 2.1 above, the mixnet encryption, ME, of each message, m, would be:

$$ME(m) = E_{p_1}(E_{p_2}(E_{p_3}(E_{p_4}(m)))) \tag{2.2}$$

The mixnet performs the decryption operation with the private key, k, of each node:

$$m = D_{k_1}(D_{k_2}(D_{k_3}(D_{k_4}(ME(m))))) \tag{2.3}$$

What we have shown so far is known as a *chaumian* or *decryption* mixnet, and any asymmetric encryption algorithm can be used in a decryption mixnet. The minimum requirements of an algorithm to create a mixnet are that the inputs be untraceable to (or different from) the output at each node. For example, *reencryption* mixnets differ from *chaumian* mixnets by partially decrypting and reencrypting each message for the next node. These mixnets can tolerate the failure of a threshold of nodes, and each node will produce a unique output even with the same input.

To make a mixnet robust, or verifiable, a technique such as *randomized partial checking* (RPC) [82] can be used. In RPC and other techniques, parts, but not all, of the mixing and decryption of the mixnet are published after they were performed so that an observer can have a high confidence that messages were not changed during the mixing process. Figure 2.2 shows RPC in action.

Note that verifying a mixnet through RPC or another method can reduce the untraceability that the mixnet provides. For example, in the mixnet without RPC, only one node needed to keep its operations secret in order to maintain privacy. In the RPC version, two nodes are needed. We can correct this problem by having each node perform two mixes.

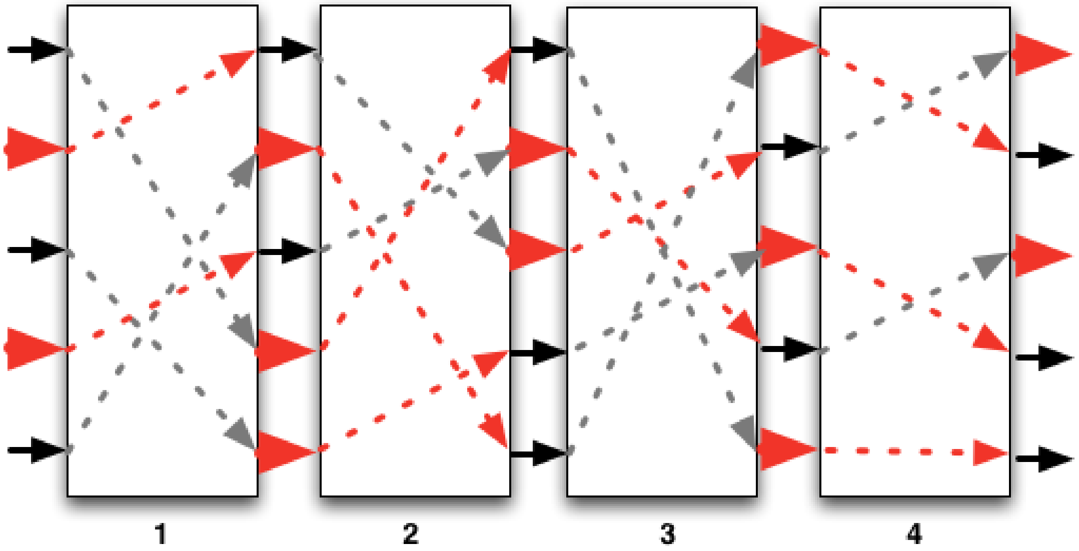


Figure 2.2: **Mixnet with Randomized Partial Checking (RPC)**. Mixnets with RPC operate in the same manner as a normal mixnet. However, after mixing is done it reveals part of the data. The second and fourth messages in this illustration are checked or audited, and this choice cascades through the mixnet, as the three other messages not chosen for the first node are chosen to be audited for the next node.

2.5 Implementations and Studies of Vote Verification Systems

There is a large body of knowledge about usability of computer security systems (see Cranor and Garfinkel [50] for an overview). There is also more specific work regarding election system usability.

Byrne *et al.* [22] experimentally compared the usability of punch cards, lever machines, and paper ballots. He found that voters made fewer errors with paper ballots. Laskowski [93] offers practical metrics for voting system usability; draft voluntary guidelines [140] also address election system usability.

Examining social issues, Newkirk [103] found that public opinion remained remarkably stable between 2004 and 2008. During that time, DRE systems were the top-rated systems for voter trust, followed closely by PCOS systems. Voters rated vote-by-mail, central count optical scan, and internet voting as less trustworthy.

Norris [106] performed a telephone survey of registered voters in Maryland in which voters provided strongly positive opinions about the usability and accuracy of touch-screen voting. Voters were also positive about the reliability, trustworthiness, and count-accuracy of touch-screen machines, while admitting that the systems could be corrupted by malware.

Public confidence in elections was rated highly in the Newkirk and Norris studies, second only to banks. More confidence was voiced for elections than medical providers (including hospitals and clinics), universities and schools, large corporations, and the government. Given the impact of public opinion on the decisions of policymakers who purchase voting systems and oversee other matters related to the administration of elections, it is important to study public reactions to voting systems. This is particularly true for E2E systems, which change the user experience by providing receipts and change voter expectations by enabling additional verifiability.

Using expert review, laboratory studies, and a field experiment with 1,540 participants, Herrnson *et al.* [76, 75, 10, 48] have found that voting system interface and ballot styles had an impact on voter satisfaction, the need for help, and voters' abilities to cast their ballots as intended. They also demonstrated that the most frequent error made by voters was voting for a candidate other than the one they intended to support, usually a candidate listed on the ballot immediately before or after the intended candidate. This type of error is more serious than the errors associated with the residual vote because, in addition to denying an intended candidate a vote, it gives a vote to one of that candidate's opponents. Results of this experiment varied by voter demographics and voting experience. They also found that design issues and voter backgrounds influence not only the voters' evaluations of different voting systems, but their voting accuracy.

To date, E2E voting systems have not been used in many elections, and the practicality and usability of E2E election systems have not been well studied. The focus of many of the existing E2E studies was on practical implementation and engineering decisions. In the following sections we examine studies of two other E2E voting system implementations: Helios and Prêt à Voter. Afterward we discuss the study of an Internet voting system with verification features, Rijnland Internet Election System (RIES). Then we discuss the studies of DREs with VVPATs and other verification equipment, and conclude with an overview of the studies of Scantegrity's ancestor system Punchscan.

2.5.1 Helios

Helios is an E2E Internet voting system implementation originally created by Adida [4]. It is based on work by Benaloh [11].

In Helios, a voter uses her Web browser to make her selections and is then presented with an encrypted ballot. The voter can then choose to audit and spoil, or cast her ballot. If cast, the encrypted ballot is posted on the bulletin board provided by the Helios system, and all the ballots are decrypted by a Sako-Killian mixnet [130]. The system does not attempt to protect against coercion. It also relies on the security of the user's computer, as a malicious workstation could pretend to audit the ballot correctly when it is really malformed to vote for another candidate [59].

In March 2009, Helios was used to elect the President of the Université catholique de Louvain [5], and again in October 2009 for the Princeton University undergraduate student government election. Used in elections where over 10,000 votes have been cast, it is credited with the largest E2E elections to date.

The designers did not attempt to get feedback from voters, but did look at some voter behavior. Among other things, they found that sending batches of reminder e-mails spiked voting activity, that very few voters used features like voting in the polling site or voting more than once, and that almost 30% of voters checked their ballots on the Web bulletin board.

They did not gauge reaction from voters and election officials. Measuring the impact on these users is critical to determining feasibility of E2E voting as an acceptable paradigm for elections, which is what we did through surveys and observations in our use of Scantegrity.

2.5.2 Prêt à Voter

In 2005, Ryan and Chaum proposed Prêt à Voter as an improvement on Chaum's earlier scheme by using a preprinted ballot [43]. In it, candidate names are printed in a vertical random permutation and an encryption of the candidate list, called the onion, is printed at the bottom of the ballot. A voter votes by marking next to the desired candidate on the list and destroying the candidate list (the left half). The surviving part of the ballot with marked positions (the right half) is scanned and taken home by the voter to be checked online.

The position and the onion are used in a mixnet to decrypt the Prêt à Voter ballot. The onion is small, and could also be a hash of the encrypted list, limiting the ability to embed a subliminal channel into the printed ballot. Some of the ballots created by the system must be spoiled and audited in order to check that the ballots are being printed correctly.

Scantegrity is derived from Punchscan, which is a close relative of Prêt à Voter but has several key differences. Punchscan and Scantegrity do not use a mixnet. The structure created is very similar but allows for much faster computation of results. The Punchscan ballot does not have an onion and uses a different format from Prêt à Voter.

Prêt à Voter was implemented and tested for the 2007 VoComp competition [13]. The developers found that there was a large gap between theoretical voting system features and the actual practice of implementing the features to meet the assumptions that are made in the designs [16]. They also found, among many other lessons learned, that requirements when implementing new voting systems are often hard to specify and tend to shift during development. We have also found both of these observations to be the case in our development of Punchscan and Scantegrity.

Prêt à Voter was also tested in an observational study of about 70 voters [143], which found overall that the system is comparable but performs slightly less well than other election systems in terms of usability, user acceptance, and trust. This trial is very similar to the mock election discussed in Chapter 4, but it was not set up to imitate an election environment. Like the mock election, the participants were not representative of the general voting population in the area.

Note that neither Helios nor Prêt à Voter had multiple ballot styles, the users of the systems were not a broad cross-segment of the population as in Takoma Park, the system implementors were typically involved in administering the equipment, and no active auditors were established to audit the elections. To date, our study of Scantegrity is the most comparable real-world use case of E2E technology to that of a typical optical scan election.

2.5.3 RIES

RIES is an Internet voting system developed by water management authorities in the Netherlands [80]. It is based on an earlier proposal by Robers [127].

In RIES, an election authority creates a reference table of each possible vote in the election attached to each possible voter ID number, and publishes this information before the election. Each voter receives a secret key that she can use to generate the vote; then she can encrypt a vote and her voter ID using the public key of an anonymizer authority to whom she submits her ballot. The anonymizer decrypts and shuffles the ballots, removes each voter ID number, and posts the ballots. Afterward, anyone can verify the tally of election results.

RIES is notable because it has been used in several Netherlands elections for the water management boards from 2004 until it was stopped due to security issues in 2008 [72]. RIES is not an E2E election system because the voter does not have indisputable proof of malfeasance if the wrong ballot is cast. It suffers from the same workstation security problems of Helios, and it permits a similar way for voters to prove how they have voted in the election. It does, however, offer some ability for each voter to verify that her ballot has been counted.

The RIES system does have reasonable theoretical security and provides some reasonable protections against fraud (e.g., the ballot creation and anonymizing are done by separate authorities). This aspect has been well studied, so it is surprising to see that—four days after the source code was published in June 2008—there was a fundamental lack of secure programming and numerous practical vulnerabilities in the implementation [72]. Many of these vulnerabilities were trivial to discover, and would have caused serious issues had they been exploited during the election. This incident offers a good case study on why open-source code should be used for voting systems.

2.5.4 DRE Vote Verification Systems

In 2006, the state of Maryland commissioned a two-part study of vote verification technology of DRE systems. Part 1 [107] of the study reports the findings of a technical review of several vote verification systems, and Part 2 [74] reports on a comparative usability study of the vote verifications systems against a DRE with no verification systems.

Both studies examined four vote verification systems that would be attached to a DRE in use in Maryland at the time of the studies: a VVPAT printer (the *Diebold AccuView Printer Module*), a cryptographic receipt printer (the *VoteHere Sentinel*) that would provide E2E integrity, an independent verifier DRE (the *Scytl Pnyx.VM*), and a Voter Verified Audio

Audit Trail (VVAAT) system, provided by researchers at MIT. Each system was a prototype or mock-up provided by each election vendor or research group.

The technical review, led by Norris *et al.* and summarized by Sherman *et al.* [137], provided a rating of each system for its ability to protect election integrity and privacy, provide accessibility and reliability, and meet other practical requirements. They found that, while all the systems might offer some verification, they were all deficient in meeting one or more of the stated requirements.

Some of the deficiencies are attributed to none of the systems having been fully developed or integrated to work with the Diebold DRE, and that they would increase the complexity associated with the act of voting. Perhaps more fundamentally, however, none of the vote verification systems would have wholly positive impacts on election integrity or privacy, and would instead improve some areas while damaging others.

The usability study, led by Herrnson *et al.* and later summarized in a book chapter by Herrnson *et al.* [75, ch. 6], examined the usability of each vote verification system using expert review and a field test involving over 800 Maryland voters. As in the technical review, the researchers highlighted the added complexity of each vote verification system to the election process and the expert review noted privacy issues. The expert review also rated the VVPAT device the highest, and noted a perceived trade-off between the usability and security in election systems.

The usability study found that voters voted most accurately and required the least help on the reference system with no attached verification device. The reference DRE system also rated higher than all of the other verification systems except for the VVPAT, where it rated only slightly lower.

The researchers also note that not many voters bothered to check that their ballots had been recorded appropriately on the vote verification device, making the presence of a verification device less effective. A later study by Everett [60] confirmed this observation, and found that fewer than 50% of voters check VVPAT receipts.

Neither study recommended that the state use any of the vote verification systems. Both studies indicate that using a separate verification device increases complexity of the voting process, and we believe that this is the fundamental problem with this approach to verification. The extra work involved for the voter naturally decreases satisfaction for the voter, and the added verifiability increases voter satisfaction only when it is simple enough to be understood and straightforward to use.

2.5.5 The Punchscan Studies

Punchscan [118, 65], the ancestor of Scantegrity, has been used in trials, student elections, organizational elections, mock elections, surveys, and the 2007 VoComp International Voting System Design Competition [26]. Punchscan was used in the 2007 student elections at the Universtify of Ottawa [57] and the 2007 board elections for the Computer Professionals for Social Responsibility.[4] For a brief overview of the Punchscan system, see the next section.

The usability of the ballot marking process in any voting system can affect the accuracy of capturing voter intention. We have found that 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 [57]. In any case, Scantegrity [39] sidesteps these potential usability shortcomings by using a standard, non-randomized candidate ordering with adjacent bubbles, such as with conventional optical scan ballots. Outside of

[4] http://cpsr.org/

operator observations of the elections and student projects, Punchscan has not been studied through any published observational or comparative studies.

2.6 Punchscan Overview

Punchscan is an E2E cryptographic voting system that is a predecessor to Scantegrity. This section is excerpted from [23], and provides an overview of this system.

We begin the overview of Punchscan by explaining the voter experience, followed by an explanation of the system architecture in Section 2.6.2. We end with a discussion of the punchboard in 2.6.3, which is the mixnet structure used by Punchscan to tally ballots. Scantegrity uses a similar structure.

2.6.1 Voter Experience

The Punchscan ballot, as illustrated in Figure 2.3, is created by combining a top and bottom sheet of paper. The top sheet has letters or symbols next to candidate names and holes in it to show letters that are printed on the bottom sheet. The letters on both sheets are ordered randomly.

To vote, each voter uses a bingo dauber to mark the letter seen on the bottom sheet that is next to the candidate of her choice on the top sheet. This action creates a mark on both sheets because the bingo dauber is larger than the hole through which the letter is viewed. Afterward, the voter destroys either the top or the bottom sheet, and the surviving sheet is scanned, publicly posted, and kept by the voter as a receipt.[5] The choice of which sheet is

[5] Alternatively, the top sheet of the ballot could consist of a random ordering of the candidates, and the voter would mark next to the candidate as is done in Prêt à Voter. This change removes the need for a choice of sheet (forcing the bottom sheet to be the choice). This modification is discouraged for two reasons: We wish to preserve an ordered candidate list as may be required by state laws, and the random choice serves as an automatic audit to check the integrity of the printing process.

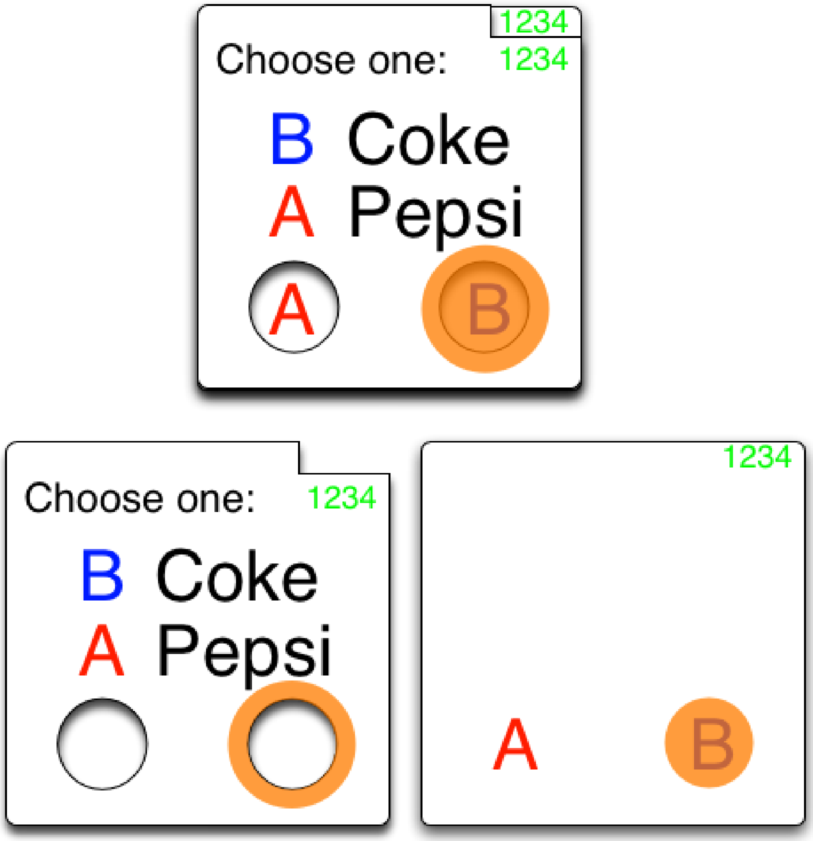


Figure 2.3: **The Punchscan Ballot**. On top, a marked ballot. On the top corner the serial number is listed twice, once on each sheet. In the center is the ordered candidate list, and to the left is a random ordering of symbols (A or B in this case). Below the candidate list, printed on the bottom sheet, is another independent ordering of the same symbols. Underneath we see that both sheets are marked after the top and bottom sheets have been separated. A top or bottom sheet by itself does not reveal any useful information about how the voter has voted.

destroyed is determined at random before the voter views the ballot. This choice can be made in a variety of ways: For example, we could have a jar with an equal number of tokens inscribed with “top” and “bottom” that are then chosen by or in front of the voter.

As shown in Figure 2.3, neither half of the ballot can reveal the original vote by itself. Only the trustees can determine the original intent using the punchboard explained in Section 2.6.3.

Also note that the position marked by the voter is known as the *mark position*, and in subsequent diagrams is either 0 for the left mark or 1 for the right mark. For races with more than two candidates, we would indicate the choice as 0, 1, ..., or n, with numbering starting at the leftmost position.

In implementation, the ballot could be represented as a full permutation or by a cyclic shift. The cyclic shift uses modulo arithmetic, and the vote position can be added to the two cyclic shifts in order to determine who is chosen. In the simple case, one bit may be used for both candidate orderings on each ballot sheet (0 for AB and 1 for BA). Representing arbitrary permutations requires more information. As shown in Figure 2.5, we can create four combinations of different ballot types.

2.6.2 System Architecture

The Punchscan architecture is illustrated in Figure 2.4. A Web server acts as the central repository for all election data. The trustees use a trusted workstation to create the ballot images, generate the election data, and create commitments to the generated election data that are posted on the Web server. Later, the election data is challenged by an auditor, and the trustees must post the information necessary to reveal the commitment data. Anyone can verify the proper disclosure of the challenged data.

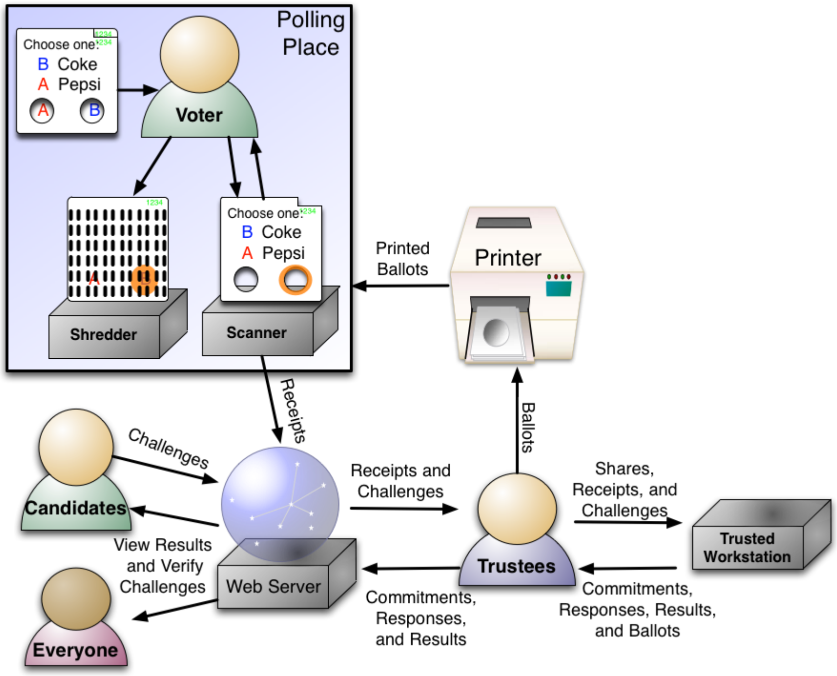


Figure 2.4: **Punchscan Architecture**. An architecture diagram of the Punchscan system. The Web server acts as a central, transparent repository for all election data. Anyone can check that the data are correct.

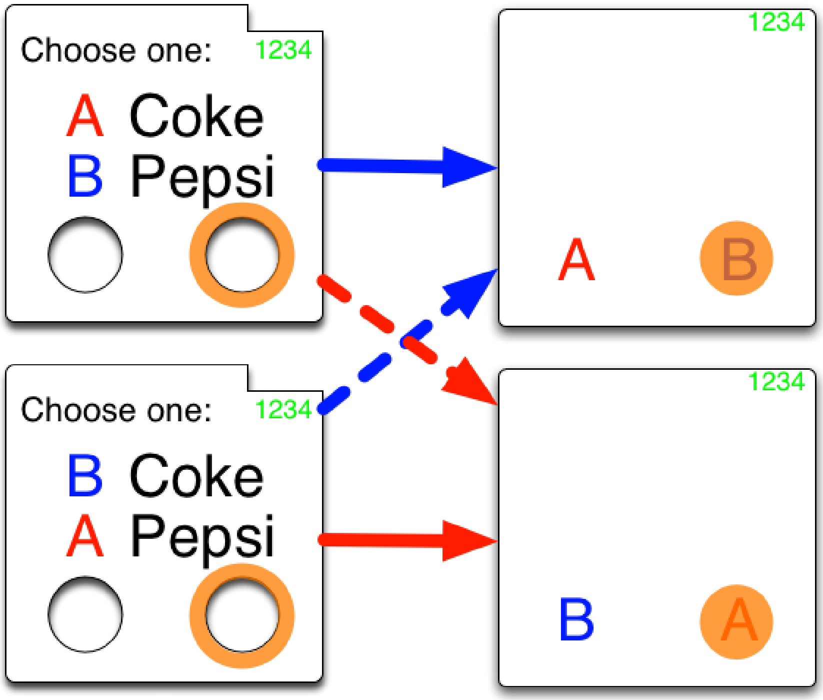


Figure 2.5: **Possible Ballot Combinations**. All possible combinations of a two-candidate ballot. Notice that while there are 4 combinations, only 2 of them are unique when combined with a vote position.

Trustees also use a printer to print the Punchscan ballot images and send them to polling places to give to each voter on election day. Each voter receives a privacy-preserving ballot receipt after voting and is able to verify that the receipt is posted on the Web server. Anyone can check a ballot, view the results, and verify the audit data posted on the Web server.

In cyclic shifting when adding (mod 2) the sheets together, adding the mark position produces the final vote. More specifically, if both the top and bottom are 0 or 1 ($0 + 0 = 0$ and $1 + 1 = 0$), then the 0 position chooses the first candidate, and 1 chooses the second, and if the top and bottom differ ($1 + 0 = 1$ and $0 + 1 = 0$), then 0 chooses the second candidate, and 1 chooses the first (just as it is seen by the voter in Figure 2.3). This generalizes to N candidates with mod N arithmetic.

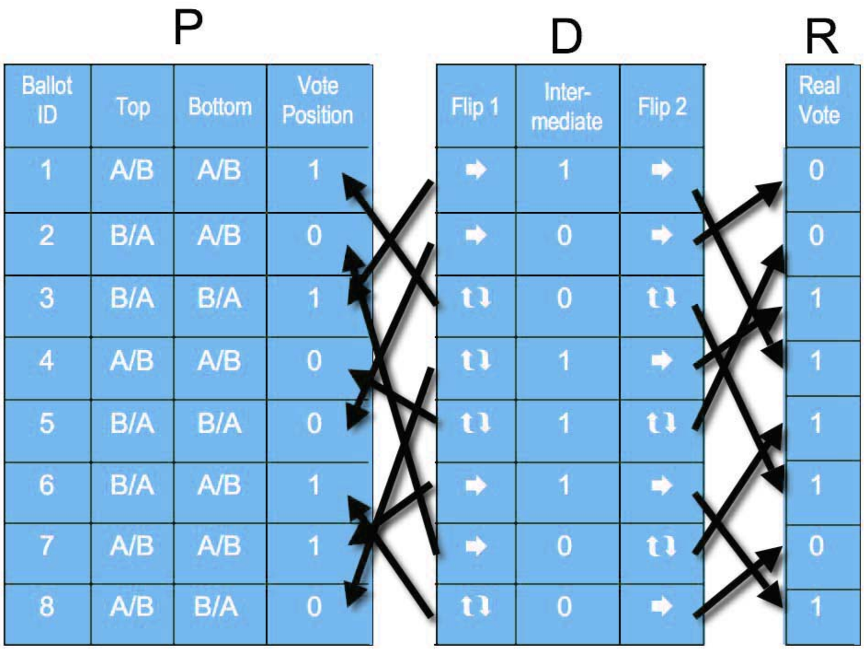
In the cyclic case, the punchboard merely takes as input the position and performs cyclic operations to turn the ballot back into canonical form (where the shift turns to 0); the number left over was the choice made by the voter. A consequence of cyclic shifting is that it can reveal information in elections where multiple choices are being made, such as in m out of n elections or ranking methods like instant runoff voting, because the spacing between the candidate lists and the choices would indicate certain unique voting choices.

If we use a permutation, the punchboard performs operations that reverse the ballot permutation, and the position is changed during this process. In the implementation it does not matter what sheet is taken as a receipt, as it is the combination of both receipts that is proved by the punchboard, and not the individual sheets.

2.6.3 Punchboard

In order to determine voter intent, the trustees must know the letter ordering on at least the destroyed half of the ballot; this information is available through the punchboard, shown in Figure 2.6. To interpret the results, candidate order is associated with a marked position in the Results (R) table Thus, a 0 position in the R table represents a vote for the first candidate listed on the ballot (Coke).

We use the punchboard to provide voter privacy and election integrity. If we post it as shown in Figure 2.6, there is no privacy in the system, but if it remains secret, we provide no publicly-verifiable integrity to the counting process. In order to achieve both of these properties, Punchscan uses bit commitment to commit to certain data before ballots are printed for the election, and RPC auditing is used to reveal parts of the punchboard as the election progresses. The punchboard performs an operation, reveals an intermediate result, and then performs a second operation to reveal a final result. Either the intermediate operation or the final operation is revealed. This method enforces integrity by making public



P

Ballot ID	Top	Bottom	Vote Position
1	A/B	A/B	1
2	B/A	A/B	0
3	B/A	B/A	1
4	A/B	A/B	0
5	B/A	B/A	0
6	B/A	A/B	1
7	A/B	A/B	1
8	A/B	B/A	0

D

Flip 1	Inter-mediate	Flip 2
➡	1	➡
➡	0	➡
↻	0	↻
↻	1	➡
↻	1	↻
➡	1	➡
➡	0	↻
↻	0	➡

R

Real Vote
0
0
1
1
1
1
0
1

Figure 2.6: **The unredacted punchboard**. Coded votes are displayed on the left side in the Print (P) table and uncoded, canonical votes are in the Results (R) table. The Flip columns in the Decode (D) table contain either a straight arrow, which leaves the vote position mark alone, or a circular arrow that flips a 0 to 1 and a 1 to 0. The top and bottom sheet columns considered together should match the Flip 1 and Flip 2 columns considered together such that 0 corresponds to the first candidate in the R table and 1 the second.

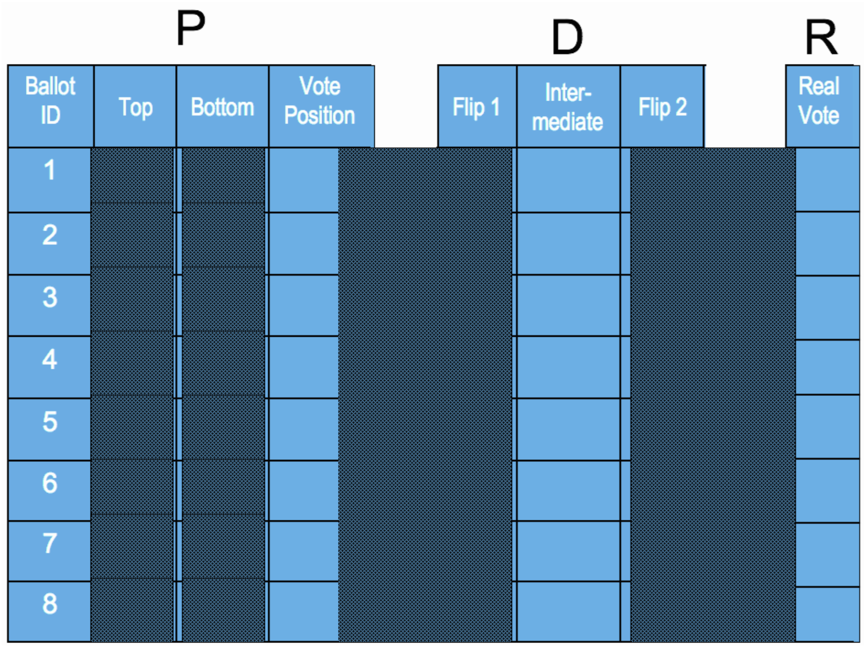


Figure 2.7: **Pre-election punchboard**. The punchboard as published before any auditing. Each cell in the P table is committed. Each Flip column in the D table, and the rows in the P and R tables to which it corresponds, are also committed.

certain values as we progress through the election after they have been committed to by the trustees, allowing anyone interested to check whether the public values, or revealed data, match what election authorities committed to before the election. The data not made public protect the privacy of voters.

In Figure 2.7, the boxes covering the table cells represent committed data. While the commitment function is the same, the data put into it and the functions of certain commitments can be split into three types, each corresponding to a different RPC operation.

The first type of commitment, the printing commitment, is a commitment of each cell for the top and bottom sheets in the P table. The printing commitment data are revealed when a ballot is spoiled, or after results are posted when the trustees know which sheet each voter took as a receipt. Thus, depending on the sheet chosen to be destroyed, the receipt not only verifies the positions chosen by the voter but also permits voters to check on the printing process.

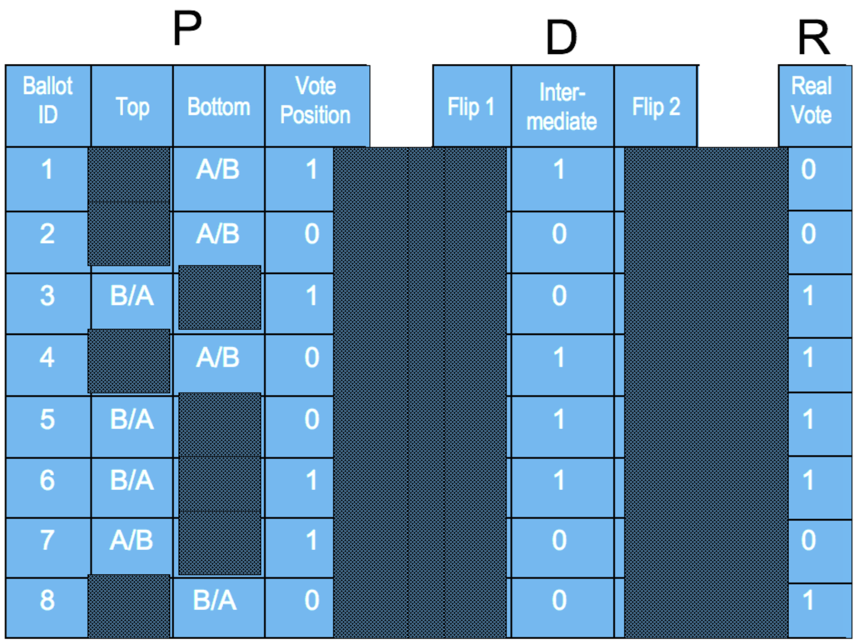
The second type of commitment is a D-row commitment, which encompasses both flips in the D table and the corresponding permutations. The data for a ballot is released only when it is spoiled. Revealing the data checks on the printing process and serves as an integrity check to make sure the flip columns correspond to the top and bottom sheet columns in the table.

The last kind of commitment, the mix commitment, consists of each of the entire flip columns (*i.e.,* Flip 1 and the permutation to the P table or Flip 2 and the permutation to the R table). After results are posted the auditor chooses which of the two commitments for each D table to reveal. This choice is a RPC operation, and allows anyone to verify that the table was filled out correctly by the Trustees, but prevents anyone from determining what rows or votes in the P table corresponded to what rows or votes in the R table.

Auditing.

There are three types of audits: pre-election, results posting, and post-election. Because a malicious person does not know what data will be chosen by the auditors, any malicious action taken has a high risk of being caught. Thus, it is important that the Trustees commit to the election data before the auditors perform any actions, because prior knowledge of intended auditor actions would let the Trustees or the attackers know what malicious actions they could take without being detected.

Pre-Election Audit. The pre-election audit ensures proper construction of the punchboard. After the trustees first publish the punchboard, auditors choose half of the rows in the P table at random and the trustees publish the contents of those rows. The published rows are checked with their commitments to ensure that they are well formed; they are

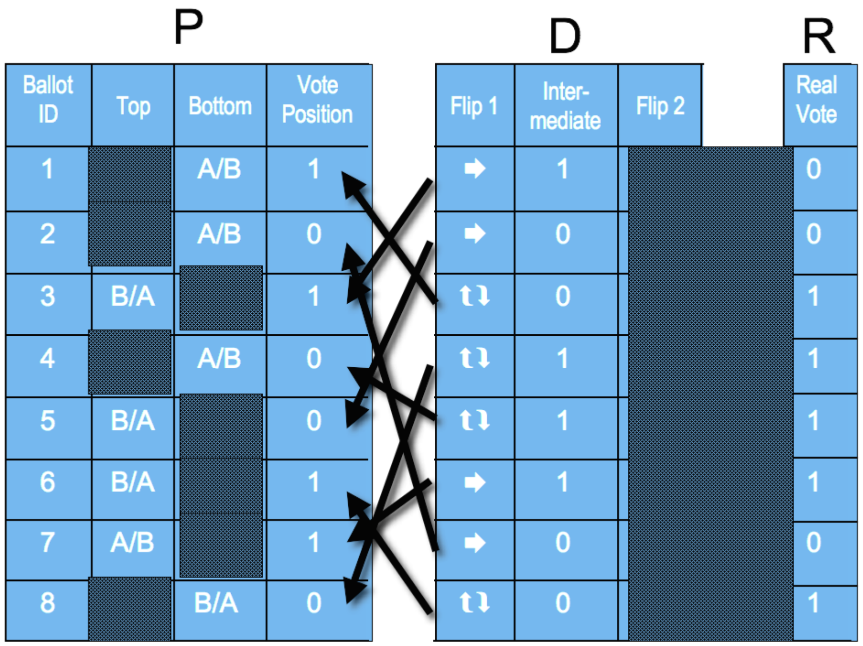


P				D			R
Ballot ID	Top	Bottom	Vote Position	Flip 1	Inter-mediate	Flip 2	Real Vote
1	▨	A/B	1	▨	1	▨	0
2	▨	A/B	0	▨	0	▨	0
3	B/A	▨	1	▨	0	▨	1
4	▨	A/B	0	▨	1	▨	1
5	B/A	▨	0	▨	1	▨	1
6	B/A	▨	1	▨	1	▨	1
7	A/B	▨	1	▨	0	▨	0
8	▨	B/A	0	▨	0	▨	1

Figure 2.8: **Post-election punchboard**. The punchboard after results are posted. The committed data for the receipts are revealed, and voters can check the punchboard to ensure their vote made it to the final tally.

then discarded and the remaining rows are used to print the sheets that make up each ballot used in the election. A sufficient number of the printed ballots should also be audited and spoiled in the same way to check on the printing process. This audit makes half of the rows unusable, so the trustees must generate more than twice as many rows as the number of needed ballots.

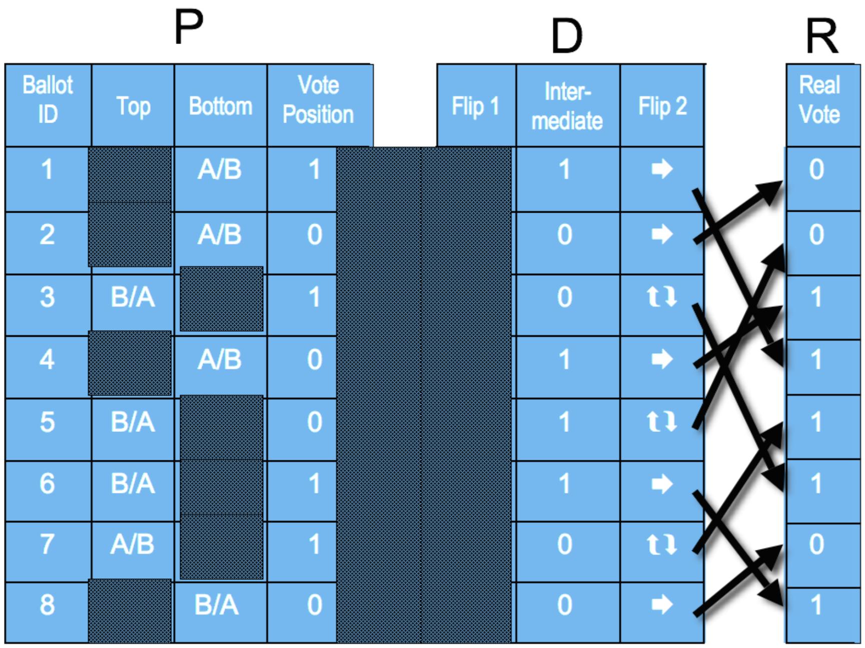
Posting Results. When results are posted, the trustees populate the Vote Position, Intermediate Position, and Real Vote columns of the tables. They additionally reveal the sheet that each voter took home as a receipt. Each voter is able to verify that her ballot was included with the correct marks in the final tally, that her receipt matches the revealed data, and that it was well-formed. Everyone is able to verify that revealed data matches what was committed to before the election. An illustration of the punchboard after the results are posted is shown in Figure 2.8.



Ballot ID	Top	Bottom	Vote Position
1		A/B	1
2		A/B	0
3	B/A		1
4		A/B	0
5	B/A		0
6	B/A		1
7	A/B		1
8		B/A	0

Flip 1	Inter-mediate	Flip 2	Real Vote
➡	1		0
➡	0		0
⇅	0		1
⇅	1		1
⇅	1		1
➡	1		1
➡	0		0
⇅	0		1

Figure 2.9: **Post-election audited punchboard**. A final version of the punchboard after the post-election audit. In this version, the auditor chooses the left side of the D table and the trustees reveal it. Now, we can see how the Vote Position column corresponds to the Intermediate Position column and verify that every vote was accurately recorded in the Intermediate Position Column.



Ballot ID	Top	Bottom	Vote Position	Flip 1	Inter-mediate	Flip 2	Real Vote
1	▨	A/B	1	▨	1	➡	0
2	▨	A/B	0	▨	0	➡	0
3	B/A	▨	1	▨	0	⇅	1
4	▨	A/B	0	▨	1	➡	1
5	B/A	▨	0	▨	1	⇅	1
6	B/A	▨	1	▨	1	➡	1
7	A/B	▨	1	▨	0	⇅	0
8	▨	B/A	0	▨	0	➡	1

Figure 2.10: **Alternative post-election audited punchboard**. A final version of the punchboard after the post-election audit. In this version, the auditor chooses the right side of the D table and the trustees reveal it. Now, we can see how the Intermediate Position column corresponds to the Real Vote column and verify that every vote was accurately recorded from the Intermediate to the Real Vote column.

Post-Election Audit. The post-election audit ensures that the counting process was executed properly while maintaining voter privacy. For each published D table, auditors choose the two columns left or right of the Intermediate Position column and the trustees reveals that data. That way, everyone can then check that the marked positions match the intermediary values, or that the intermediary values match the final results. Because the trustees or attackers did not know which half of each D table would be selected before they populate the Intermediate Position and Real Vote columns, improperly publishing a result in either column would result in an overwhelming probability of being caught. Illustrations of both column choices are given in Figure 2.9 and Figure 2.10.

Alternate Audit.

Instead of the entire column, the columns on each row could be individually chosen. This may be undesirable, as it reduces the privacy set by half. That is, there become two distinct groups of ballots: those whose left column was opened and those whose right column was opened. We can group the results into these two sets.

Recount or Reaudit.

If some impropriety is found in the auditing process, the post-election audit can be rerun by producing new D tables, publishing the results, and having the auditors pick from the new tables. It is not possible to rerun the first two audits because they are integrity checks on the printed ballots and voted positions. Because of the counting protocol, any counting mistakes can be detected in the same way as before.

Chapter 3

The Scantegrity Voting System

Scantegrity is an E2E verifiable voting system for bubble-style optical scan election technology. It was initially designed to be an add-on for pre-existing systems, but we were unable to get permission from a vendor to integrate with its technology. Also, there are potential technical issues introduced by the verification mechanism, so we built our own underlying optical scan system implementation designed to support it.

The E2E functionality of Scantegrity is enabled by the use of confirmation numbers. In the system each voter uses a special pen to mark his or her ballot. The pen makes legible pre-printed confirmation numbers corresponding to each ballot selection. The link between confirmation numbers and the selections is cryptographically protected, with the key(s) being secret shared by election officials. Each voter can copy the confirmation numbers to a chit that is detachable from the ballot. After the election, all voters can verify that their confirmation numbers are included unmodified in the collection of confirmation numbers posted as a public record; additionally, anyone may verify that the tally is computed correctly from the same public record.

To verify that the correspondence of each confirmation number to each candidate is correct, all voters and authorized observers may "audit" ballots by requiring the system to expose all confirmation numbers and corresponding selections on the audited ballots, and by checking that these correspond to those printed on the ballots. The audited ballots are not used for voting. The central election authority computes the final tally in a verifiable manner from the posted confirmation numbers.

The verifiability property of Scantegrity is independent of voting system software correctness and ballot chain-of-custody after ballots are cast. The proof of correctness made by those running an election is based only on (a) the inability of the voting system to change values once they have been committed to a public record, and (b) the unpredictability of choices made by voters and election auditors—to verify confirmation numbers online, to audit ballots, and to audit the data provided by the voting system regarding the processing of confirmation codes to obtain the tally. Election officials can also use the system to expose false charges of election fraud.

Because Scantegrity is an overlay on paper ballot systems, it cannot remove the inherent ballot secrecy limitations of the underlying paper ballot election system, some of which are highlighted in work by Jones [83]. For example, a voter can be identified by a distinctive manner of making marks, or the order of voters scanning ballots can be analyzed to determine each voter's selections. Also, the miniaturization of cameras poses challenges to the secrecy of voter selections in all types of voting systems.

Scantegrity does, however, attempt to limit any additional ballot secrecy vulnerabilities. For example, the linking of confirmation numbers to votes requires the collusion of a set of election officials or the breaking of the security of the cryptographic functions. Furthermore, the use of a slow-reacting ink and a modification to the voting procedure can ensure that information linking confirmation numbers and ballot serial numbers to voter selections can be removed from ballots a few minutes after they are marked.

As with regular optical scan, forensic and machine attacks are possible. Coercive adversaries could, for example, use specialized equipment to attempt to read the codes on the ballots or compromise the scanner. We assume these attacks are too time-consuming and unwieldy to be very practical, however, for two reasons: First, we have instituted printing procedures to minimize the effectiveness of such ballot analyses; these are described in Section 3.4.1. Second, simpler attacks, based on fingerprinting the underlying paper using commodity scanners [46], are possible against paper-ballot-based voting systems.

This chapter contains an expanded version of the Scantegrity description found in [40]. Section 3.1 gives a procedural non-technical overview of the system, and Section 3.2 provides the formal technical description. Section 3.3 provides a security analysis. Implementation details, which are not provided in [40], are given in Section 3.4.

3.1 Scantegrity Procedures

The procedures define the manner in which participants in the election—voters, election administrators, and observers—interact with the voting system in order to ensure that:

1. Confirmation numbers are present and correct on the ballots.

2. Marked confirmation numbers are present and correct on the website.

3. Confirmation numbers are processed correctly to obtain the final tally.

The procedures are designed to enable the detection of election fraud if it has occurred, as well as to prevent false charges of election fraud. This section provides an informal description of a more formal protocol; its purpose is to provide a description that is somewhat accessible to voters, election judges, and election administrators, and to prepare the reader for the more formal description in the next section.

3.1.1 The Vote Casting Procedure

The vote casting procedure is very similar to that of a regular optical scan ballot. The slight differences between the two are as follows:

1. The unmarked ballot itself looks slightly different: it bears a detachable chit with hidden serial numbers that is used to record and check confirmation numbers.

2. While marking the ballot, voters will notice the appearance of confirmation numbers, which will also disappear after a few minutes.

3. Voters or observers may *audit* ballots to determine whether printed confirmation numbers correctly reflect voter selections; such ballots may not then be cast.

4. Voters interact with a polling official after the vote is successfully cast to expose serial numbers on the receipt chit.

While we have simplified the ballot audit procedure considerably from previous proposals, it does not have a corresponding equivalent in the regular optical scan protocol and might appear complicated to voters and officials. Similarly, spoiled ballots are discarded using a procedure that is similar to (but more complex than) that used for optical scan.

The Ballot

The Scantegrity ballot consists of two parts: the *main body* and the *chit*; see Figure 3.1. As with an optical scan ballot, the main body of a Scantegrity ballot contains, for each contest, a list of valid selections printed in a canonical order predetermined by polling place

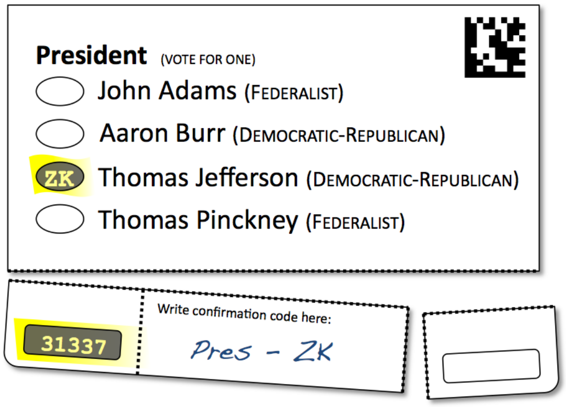

President (VOTE FOR ONE)
John Adams (FEDERALIST)
Aaron Burr (DEMOCRATIC-REPUBLICAN)
ZK
Thomas Jefferson (DEMOCRATIC-REPUBLICAN)
Thomas Pinckney (FEDERALIST)
31337
Write confirmation code here:
Pres - ZK

Figure 3.1: A Scantegrity ballot showing the main body (top) with one marked position and machine-readable serial number; left chit (bottom left) with a developed chit serial number and confirmation number written in; and right chit (bottom right) with an undeveloped chit serial number. This figure is meant to illustrate the parts of the ballot and does not represent the actual final state of the portions after voting, which show both chit serial numbers for a cast ballot.

procedures (e.g.,alphabetical, rotated across precincts, *etc.*). Next to each possible selection is a markable region, oval in shape.

Differing from an optical scan ballot, the background of each oval is printed with a reactive ink. The confirmation number corresponding to the selection for the particular ballot is printed inside the oval. The ink used to print the confirmation number is similar to that used for the oval background, but is slow reacting. Thus, before marking, the oval has a single color and confirmation numbers are indistinguishable from the background of the oval (*i.e.,* invisible). Additionally, a Scantegrity ballot contains a ballot serial number that is machine readable, but not easily read or memorized by a human (e.g., a two-dimensional barcode).

We assume that both inks are indistinguishable to the human eye before the oval is marked with the ballot-marking pen (See Section 3.4.1 for details on the validity of this assumption). We also assume that voters will not be able to take analysis equipment into the polling booth, as such equipment might be able to distinguish between background and confirmation number.

The chit is attached to the bottom of the ballot via a perforation so that it can be easily detached. It has two halves, left and right; the halves can be detached from each other using a pair of scissors. On each half is a chit serial number: the left chit serial number and the right chit serial number. These chit serial numbers are distinct from each other and from the ballot serial number; we describe later how they are used to ensure that voters cannot make false claims regarding confirmation numbers on uncast ballots. Both the left and right chit serial numbers are printed in invisible ink so that they are neither human nor machine readable before being decoded using a special decoder pen. Both the left and right chit serial numbers are assumed to appear after they are marked with the decoder pen.

Ballot Marking

Upon arrival, a voter is authorized to cast a ballot, and is handed the next ballot in the pile; it is enclosed in a privacy sleeve. At this time, she may choose to audit a ballot, which she may choose from the existing ballot pile. For details on the ballot audit procedure, see Section 3.1.2.

In order to select a candidate, the voter fills in the corresponding oval using a ballot-marking pen. In accordance with the invisible ink printed on the ballot, the background of the oval will immediately turn dark, leaving a confirmation number visible in the foreground. The relative darkness of any marked ovals to unmarked ones will allow an optical scanner employing dark mark logic to register the oval as marked.[1] The foreground of the oval will be human readable, and a voter interested in verifying that her vote is in the digital collection

[1] The vendors that have seen this system point out that this will cause problems with their technologies, which are sensitive and may read only completely filled out bubbles on the ballot.

of votes to be tallied may record the code on the chit portion of the ballot. Uninterested voters can ignore the codes.

The link between a confirmation number and the corresponding selection on a particular ballot is protected cryptographically. We explain the details underlying the generation and protection of the data in the next section. At this stage, however, we note the following: The disclosure of a confirmation number does not reveal the selection if the cryptographic techniques used are assumed secure and election officials do not to collude to determine the selection.

Although not apparent to the voter, the confirmation number is printed in a slow-reacting invisible ink that will also turn dark, but only after the passage of several minutes (e.g., five to seven minutes). At this time, the oval will be completely dark and the code will no longer be visible, leaving no human-readable unique information on the ballot.

As an option, the two-dimensional bar-coded serial number could also have slow-reacting ink in its background so that if a voter marked it, it would turn solid black.

Section 3.4.1 describes how a masking ink and appropriate printing techniques may be used to reduce the ability to distinguish between the inks, even with the use of microscopes and spectral equipment. Indeed, it may be assumed that the slow- and fast-reacting invisible inks are, for all practical purposes, indistinguishable (1) before exposure and (2) within T seconds after both have been exposed, where T is the response time of the slow-reacting ink. After a period long enough to include reaction times, a filled-in Scantegrity ballot provides, for all practical purposes, an amount of information that is similar to that on an optical scan ballot, and can be used in a manual recount with a level of privacy very similar to that of optical scan.

Spoiling Ballots

If the voter makes an error in marking a ballot, it is returned to the poll worker. Without seeing the contents of the ballot, the election judge removes the ballot from the privacy sleeve and detaches the right side of the chit from the ballot. The main body and left chit are shredded in view of the voter. The right chit is retained by the election judge and used to verify that the number of ballots issued is identical to the sum of the number of ballots tallied, print-audited, and spoiled. The number of spoiled ballots allowed per voter is typically limited by predetermined polling place procedures.

Casting a Ballot

When the voter has satisfactorily marked a ballot, it is returned to the poll worker. As previously, the election judge detaches the chit from the ballot. Further, with the choices on the ballot still concealed, the election judge places the main body of the ballot into the scanner, which records the ballot serial number and the marked choices.

In the preferred version of the protocol, voters are not allowed to cast undervoted or overvoted ballots. If a voter does not wish to vote for a particular candidate, she must make a selection of "none of the above." If the scanner detects an undervote or overvote, the voter is returned her ballot, and will spoil it and re-enter the issuance procedure.

Note that in the United States the requirement that a voter be notified of undervotes or overvotes is not uncommon; in fact the Help America Vote Act [1] requires that voters be notified of overvotes if electronic equipment is used. However, requiring that undervoted or overvoted ballots not be cast is considerably stricter, and decreases the usability of the voting system.

An alternative version of this protocol would be not to ban undervotes or overvotes in cast ballots. However, in this version, a secure chain of custody is required to ensure that unvoted races were not changed to voted ones, or that voted races were overvoted. Our goal, however, is to avoid any reliance on chain of custody or trusted components in the polling place.

After a successful scan, both human readable serial numbers on the chit are developed by the election judge. The voter may leave with the chit. It is expected that public interest groups will make available the possibility of creating a copy of chits to alleviate the need for concerned but time-constrained voters to participate in auditing the election personally.

Casting without Automation

For polling places without adequate voting technology, or in the event of a power failure, Scantegrity can still proceed with the voter being issued the chit in the same manner. The main body of the ballot would, instead of being scanned, be placed into a sealed ballot box that has been certified as being empty prior to sealing. If scanning technology is unavailable at the polling place, the ballots are then transported to a central scanning location.

Voters With Disabilities

Some voters have visual or motor disabilities, and hence cannot mark a paper ballot. These enhancements to the voting process are inspired by those for Punchscan and Prêt à Voter described in [42]. The voter is presented with an audio ballot and interacts with the voting system using a microphone and headphones. The voting system prints the vote on a Scantegrity ballot. The voter with visual disabilities also has access to a trusted interactive device that translates a visual signal into another type of signal, such as an audio signal; this device is used to check a marked ballot. Finally, all voters using the audio interface have access to a personal voice recorder used to record the confirmation number.

Filling a Ballot: The voter is presented the choices for each race through the headphones, and communicates her choice to the voting system through the microphone. The voting system communicates the vote to a printer. The printer prints, on a Scantegrity ballot, with the ink used in the Scantegrity pen, a blob on the corresponding oval, exposing the code as with ballots for other voters. Assistive devices that have been used in the past to help voters with visual or motor disabilities may also be modified for the purpose of filling a Scantegrity ballot. Examples of such devices include Tactile Ballots which have been used in elections in Rhode Island [69], and the Voting-on-Paper Assistive Devices (Vote-PADs),[2] which consist of a plastic ballot-sleeve, tactile indicators, and an audio tape recording, customized for each election and ballot design.

Checking a Marked Ballot for Correctness: A voter who has motor disabilities that make it difficult to mark ballots may check the correctness of the filled ballot, and dictate the confirmation number into a personal (trusted) voice recorder.

The voter with visual disabilities will use a trusted interactive device—consisting of a trusted scanner with optical character recognition (OCR) and speakers—to check that the ballot is correctly marked. The voter may bring such a device with him or her to the booth, or may be provided one by a trusted third party, such as a public interest group. With the aid of this device, the voter may translate a marked ballot into an audio signal, and determine if it has been correctly marked. Additionally, this device would read aloud the confirmation number, which could be taped into a personal (trusted) voice recorder.

Casting the Ballot: Once a ballot has been correctly marked, it may be processed like any other marked ballot.

[2] Accessible voting without computers, see http://www.vote-pad.us/.

Effects on Security Properties: This approach does not provide the same security guarantees to voters with visual disabilities as those provided to other voters, who need not rely on a trusted device in the polling booth. A compromise of the trusted device can result in a compromise of the integrity of the vote, as well as in an opportunity for a coercive adversary. On the other hand, the only implemented voting systems that are usable by voters with visual disabilities—DREs, optical scan ballot systems with assistive devices, or Prime III [53]—require that the voter either trust the voting system itself or the chain of custody on the ballot box or a paper/audio audit trail. These are stronger assumptions than the requirement that a *personal* device be trusted by a voter. The voter may still determine that her vote is among those tallied, and that the collection of votes is tallied correctly—without having to trust a device provided by election officials.

Accounting for Ballots

At the end of the day, election judges and official observers take note of the numbers of spoiled, voted, and audited ballots. They compare this sum to the number of used ballots and make these numbers publicly available. This allows detection of ballot stuffing. Furthermore, observers can note the exposed chit serial numbers of voted, spoiled and audited ballots so they cannot be changed after the election.

3.1.2 Election Audit Procedures

A voter may participate in auditing the election in several ways. In addition to checking the confirmation numbers on her ballot, she may audit a printed ballot, and check the processing of confirmation numbers. Election observers may also participate in the latter processing check.

Auditing a Printed Ballot

Voters wishing to audit a printed ballot may choose one from the ballot pile; we refer to the process of auditing the ballot as the *print audit*. They are each a issued a ballot main body and the left or right half of the chit, with the serial number activated using the decoder pen; which half is chosen may be determined by a flipped coin. The other half of the chit is removed and retained by the poll worker in a clear box on the election judge table. At her leisure, the voter fully marks the ballot to reveal all the confirmation numbers, which she may check using the procedure in the following section.

Checking Confirmation Numbers

At a prearranged time after the polls close, voters who recorded the confirmation numbers associated with the candidates they voted for, or those who wish to check the confirmation numbers on a print-audited ballot, may visit a website where they can access this information via the serial number on the chit.

In the case of voted ballots, the voter will have two serial numbers, left and right; either is suitable to identify the ballot uniquely. Upon lookup of a serial number, the record will show the confirmation numbers in the positions the system believes were marked for voted ballots, but will not report the candidates associated with these codes. For this reason, providing a copy of the confirmation numbers in no way undermines the secrecy of the ballot. Voters are able to share their confirmation numbers, share photographs of their chits, or post screen captures of the results.

In the case of an audited ballot, the serial number will show all the confirmation numbers that should appear on the ballot and, only in this case, also reveal the candidates associated with each code.

All confirmation numbers and their associated candidates are committed prior to the election to ensure the values or associations cannot be changed. Thus, the audited ballots provide probabilistic evidence that the confirmation numbers were correctly printed on the ballots. The correct and full inclusion of confirmation numbers from a voted ballot provides probabilistic evidence that the votes were properly scanned and not maliciously altered. Full details are provided in Section 3.2, and the strength of this evidence is quantified in Section 3.3.

Checking the Processing of Confirmation Numbers

Due to the commitments to confirmation numbers and candidates before the beginning of the election, it is known that candidates are mapped to confirmation numbers and that this mapping cannot be changed. Furthermore, through the print audits, voters are assured that this mapping has been faithfully transformed to the printed ballots they marked. By checking the inclusion of their confirmation numbers, they are further assured that the marks they made for candidates have been faithfully tranformed to confirmation numbers consistent with those on the ballot. The final step is to check that the confirmation numbers are properly mapped back to the correct candidates.

The protocol for achieving this check is based on an open specification. To check the processing of the confirmation numbers, voters may obtain software from a software provider they trust or write their own software. All required information for writing the software (such as the format of the data and the data) is provided to all interested parties. Those administering the election should appoint an independent auditor to perform this check to provide at least one audit of the tally computation from confirmation numbers. The details of this check are also provided in Section 3.2.

3.1.3 Dispute Resolution Process

If a voter discovers incorrect confirmation numbers or ballots that are incorrectly designated as voted, print-audited, or spoiled, he or she may file a dispute. In the case of an incorrect confirmation number, the dispute must provide the confirmation number that should be on the ballot. A voter's knowledge of a valid confirmation number on the ballot that is not present on the website suggests an error or malfeasance; the validity of the code can be established since the codes were committed to before the election, and the likelihood of guessing a correct code can be minimized the use of longer codes (exact quantification to follow in Section 3.3).

If a voted ballot is incorrectly designated, the voter can provide both chit serial numbers to prove that it was voted. Similarly, if a print-audited ballot is incorrectly designated, the voter or independent auditor can provide all the confirmation numbers on the ballot to prove that it was print-audited. If the voter knows all confirmation numbers in an overvoted ballot, this ballot's designation cannot be changed to print-audited, as the voter knows both serial numbers. In order to ensure that unvoted races are not voted and that properly-voted ballots are not changed to overvoted ones, a restriction of not allowing undervotes or overvotes on cast ballots is required.

3.2 Cryptographic Backend and Proof of Tally

This section describes the method for proving the correctness of an election outcome while maintaining voter anonymity.

3.2.1 Ballot Definition

For simplicity we consider a notation based on a single contest ballot. Generalization to ballots containing multiple races, as well as elections containing multiple ballot styles, should be viewed as multiple independent executions of the single contest case described herein. Let $L = (s_0, \ldots, s_{n-1})$ define a list of n ballot selections (e.g., candidates, choices).

3.2.2 Roles

We consider three categories of entities participating in the election with the acknowledgement that the entities are role-based; thus, an individual might possibly assume any or all roles.

Voters. Voters are people with the authority to cast a ballot in the election. We assume that voter authentication (external to this discussion) is undertaken prior to ballot issuing and that only authenticated voters are issued ballots. In this section we refer to a particular voter as V.

Election Trustees. Let T be the set of t election trustees, $T_1, \ldots, T_t \in T$. The trustees engage in the cryptographic protocol to set up and generate the correctness proofs of the election. T would generally consist of public officials and, optionally, candidate representatives. The protocol is intended to proceed when a minimum number of trustees are present and does not require the presence of all trustees to mitigate the disruption caused by any particular trustee's absence at various stages of the protocol.

Verifier. The set of verifiers A consists of all agents verifying the correctness proofs herein. The intention is that the tally-correctness be "universally verifiable" as defined by Sake and Killian [130]: any voter, citizen, or observer can participate, either directly or through delegation, in the verification of the tally.

Other Entities. Election judges are responsible for administering the voting process, instructing and assisting voters, and enforcing the registration, ballot issuing, marking and casting procedures outlined in the previous section.

Finally, we require the existence of a public bulletin board $\mathcal{BB}$, which implements an append-only public record. In practice it might be implemented as a mirrored public website.

3.2.3 Functions

In this section, we outline the main functions used in the protocol. For a positive integer len, we use $[len]$ to denote the set of integers $[0, 1, \ldots, len - 1]$.

The functions consist of:

1. A parameter initialization function that, given a security parameter, provides an election-specific nonce and minimum key lengths.

2. A trustee threshold-key generation function that produces individual trustee keys for trustees and a master key that can be reconstructed from a minimum threshold τ number of trustee keys. This function takes as input the election-specific nonce, the value of τ, and input bit strings from the trustees, the entropy of which provides the entropy of the keys generated.

3. A master-key reconstruction function that, given a set of τ or more trustee keys, reconstructs the master key.

4. A subkey generation function that is a cryptographic one-way function, accepts a master key and an identifier, and outputs another key.

5. A keyed permutation function that, given a key and the value len, generates a pseudorandom permutation of integers in the range $[len]$.

6. A cryptographic commitment function that is computationally hiding and computationally binding.
7. A ballot generation function that, given the candidate list, the confirmation number alphabet and length, the election master key, and the number of ballots required, generates the master list of ballots.

Details of each of these functions follow.

Parameter Initialization

$$P \leftarrow \mathsf{Parameters}(1^p) \tag{3.1}$$

This function accepts a security parameter p and outputs a set of functional parameters P, including a unique election-specific nonce λ selected in accordance with a public convention (not considered here) and a specification of cryptographic algorithms used to realize certain cryptographic one-way and trapdoor functions and specify their enforced minimum key lengths. For brevity, we will omit continual reference to P by assuming all following functions accept it as input.

Trustee Threshold-Key Generation

$$(k_1, \ldots, k_t, K) \leftarrow \mathsf{TrusteeKeys}(\omega_1, \ldots, \omega_t, \tau, \lambda) \tag{3.2}$$

Trustee key generation accepts an arbitrary-length random bit string, denoted $\omega_i \in \{0,1\}^*$, from each trustee T_i, as well as a threshold $1 \leq \tau \leq t$ specifying the number of trustees needed to reconstruct a unique election master key K.

The function outputs a distinct key for each of the trustees, $k_1, \ldots, k_t$, as well as a master key K. We do not consider the policy guidelines for selecting trustees or τ in this section. K is such that, if at least one ω_i is uniformly distributed across all possibilities, K will be as well. K is also dependent on the election nonce λ, so if the same value of ω_i were supplied in a different election, K would be different. K is only used as private input to other functions. Each output key k_i is transmitted over an authenticated and physically untappable channel to the corresponding trustee T_i.

Election Master Key Reconstruction

$$\emptyset / K \leftarrow \mathsf{ElectionKey}(\{j_1, \ldots, j_n\}) \tag{3.3}$$

This function accepts as input a set $\{j_1, \ldots, j_n\}$ of keys and outputs the unique election master key K if and only if $|\{j_1, \ldots, j_n\} \cap \{k_1, \ldots, k_t\}| \geq \tau$. Otherwise it returns a symbol (denoted by $\emptyset$) indicating the function failed to reconstruct the key.

We assume that, for the two preceding algorithms, given any unbounded adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ (over a random guess) in guessing K, given any set containing fewer than τ keys from $k_1, \ldots, k_t$, is exactly zero. One suitable construction is due to Pedersen [112], and has been suggested for use in voting by Benaloh [11]. A suitable notion of an untappable channel is the one due to Sako and Killian [130].

Sub-key Generation

$$\kappa_{ID} \leftarrow \mathsf{SubKey}(K, ID) \tag{3.4}$$

This is a cryptographic one-way function that accepts a master key K and identifier ID and outputs another key κ_{ID}, where ID defines what key is to be generated.

Keyed Permutation

$$\pi \leftarrow \mathsf{Perm}(\kappa, len) \tag{3.5}$$

This function accepts a key κ and list length len, and outputs $\pi : [len] \rightarrow [len]$, where π is a permutation selected pseudorandomly from the set of possible permutations of len elements Π_{len}. The function π depends on κ. We use the notation $X'(i) \leftarrow X(\pi(i))$ to denote the element-wise shuffle of a len-element set X for $0 \leq i < len$. Finally, we define a special-case null index, denoted $\emptyset$, in which $\pi(\emptyset) = \emptyset$ for all $\pi \in \Pi_{len}$.

Cryptographic Commitment

We consider a cryptographic commitment protocol as including the following pair of functions:

$$\bar{m} \leftarrow \mathsf{Commit}(\kappa, m) \tag{3.6}$$

and

$$0/1 \leftarrow \mathsf{Decommit}(\kappa', m', x) \tag{3.7}$$

Commit accepts a key κ and an arbitrary length message m to obtain a commitment $\bar{m}$. Decommit accepts a commitment x, key κ', and message m', and outputs 1 if $\mathsf{Commit}(\kappa', m') = x$. Otherwise it outputs 0. The cryptographic assumptions for these algorithms are as follows:

1. Given any probabilistic polynomial time-bounded $\mathcal{A}$ producing messages m and m' and keys κ and κ', the probability that

   $$\mathsf{Commit}(\kappa', m') = \mathsf{Commit}(\kappa, m)$$

   when $m \neq m'$ is a negligible quantity in the security parameter p. That is, $\mathcal{A}$ cannot find two distinct messages that produce the same commitment. This is an informal definition of the computationally binding property of a commitment.

2. Given any probabilistic polynomial time-bounded $\mathcal{A}$,

   $$|Pr[\mathcal{A}(\mathsf{Commit}(\kappa, m)) = 1] - Pr[\mathcal{A}(\mathsf{Commit}(\kappa, m')) = 1]|$$

   is a negligible quantity in the security parameter p. That is, $\mathcal{A}$ cannot distinguish between a commitment to m and one to m' if the commitments use the same key. This is an informal definition of the computationally hiding property of commitment functions.

Generate Ballots

$$\mathbf{P} \leftarrow \mathsf{GenerateBallots}(L, \Sigma, l, K, b) \tag{3.8}$$

Generate Ballots accepts ballot selection/candidate list L, confirmation-code alphabet Σ (typically the set of alphanumeric characters), confirmation-code length l, election master key K, and the overall number of ballots to be generated b. **P** contains three lists. The first is a list of b ballots, sorted by serial number, each with n selections, each selection associated with a confirmation number in Σ^l. In addition to this list, **P** also bears space for the voters' choices after ballots are filled and a third list that bears the corresponding candidates.

Let **P** denote the canonical "master" list associating codes, candidates, and voter-made marks, which we define as the triple of $(b \cdot n)$-element lists $\mathbf{P} = \{\mathbf{Q}, \mathbf{R}, \mathbf{S}\}$. For all $0 \leq j < bn$,

1. **Q** is a list of serial numbers and confirmation numbers, including serial numbers (α, β, γ) for each ballot, and confirmation numbers q for each selection in a ballot. Let $\mathbf{Q}(j) = \{\alpha_j, \beta_j, \gamma_j, q_j\}$,

2. **R** will eventually represent the list of scanned voter-made marks $r \in \{0, 1\}$ indicating the absence or presence of a mark (*i.e.,*vote) made for an associated selection. Let $\mathbf{R}(j) = r_j$, and let all r_j be initialized to 0,

3. **S** is a list consisting of b repetitions of selection/candidate list $L = (s_0, \ldots, s_{n-1})$. Let $\mathbf{S}(j) = s_{(j \bmod n)}$.

For notational convenience throughout this paper, we will use the index g to refer to a given ballot B_g and its associated voter receipt V_g where $g = \alpha_j = \lfloor j/b \rfloor$. For any $j \neq j'$ let $\alpha_j = \alpha_{j'}$, $\beta_j = \beta_{j'}$, $\gamma_j = \gamma_{j'}$ if $\lfloor j/n \rfloor = \lfloor j'/n \rfloor$.

Serial numbers β, γ will be selected independently (without replacement) by a secure pseudorandom number generator seeded by the election master key K. These numbers will be selected from a range defined by p, such that correctly guessing an unknown β or γ would occur with a small (but not cryptographically negligible) probability.

Finally, confirmation numbers q will be independently selected by a pseudorandom generator such that confirmation numbers are not repeated across a given ballot B_g, namely $q_j \neq q_{j'}$ if $\lfloor j/n \rfloor = \lfloor j'/n \rfloor$, for distinct j, j'.

See Figure 3.2 for an example of a list of four ballots when there are two candidates on the ballot, and confirmation numbers consist of three alphanumeric symbols.

j	α	β	γ	q
0	0000	7973	4630	7LH
1	0000	7973	4630	WT9
2	0001	2567	1490	J3K
3	0001	2567	1490	TC3
4	0002	4900	7891	9JH
5	0002	4900	7891	J3K
6	0003	1631	5275	KWK
7	0003	1631	5275	H7T

(a) Table **Q**

j	r
0	0
1	1
2	0
3	1
4	1
5	0
6	0
7	1

(b) Table **R**

j	s
0	Alice
1	Bob
2	Alice
3	Bob
4	Alice
5	Bob
6	Alice
7	Bob

(c) Table **S**

Figure 3.2: Tables **Q**, **R**, and **S** when there are two candidates, $s_0 =$ Alice and $s_1 =$ Bob. For example, a vote for Alice on Ballot 0000 would reveal the confirmation number 7LH; however, one for Bob would reveal WT9. Note that, for purposes of illustration, we show one way in which the **R** table may be populated based on votes cast during the election. The function GenerateBallots(), however, initializes all these values to zero. In this example, the votes cast on Ballots 0000, 0001, 0002 and 0003 were for Bob, Bob, Alice, and Bob respectively, and would reveal confirmation numbers WT9, TC3, 9JH, and H7T respectively. The published versions of tables **Q** and **S** will contain commitments to the information shown above, this detail is provided in Step 6c of the setup phase in section 3.2.5. There is no information in Table **R** before votes are cast, and there is no information made public about this table before the election.

3.2.4 Trusted Computation Platform

We assume the existence of a hardware device, referred to as the *trusted computation platform*, which the trustees use to evaluate the various functions described above. This device relies on the following assumptions related to the preservation of ballot secrecy:

- **Private and authenticated input**: the ability to receive input from authenticated trustees via a physically untappable channel.

- **Private evaluation**: the ability to evaluate a function so that the intermediate values cannot be recovered by passive or active attack of the hardware or software components.

- **Correctness**: the ability to attest that the functions being evaluated are equivalent to available and predefined source code.

Note that the correctness assumption enables the trustees to be certain that the required computations are being computed correctly, and hence increases the reliability of the computation from the perspective of the honest trustee. It does not affect the ability of the voter or the auditor to detect a cheating trustee.

With the failure of any of these trust assumptions, it may become possible for a malicious subset of trustees to recover information related to the association between voting intent and ballot serial number. For example, this can be accomplished by observing a sufficient number of trustee keys, observing intermediate state, or altering the functions to overtly or covertly leak this information.

None of these assumptions, including the correctness assumption, dictate the soundness of the tally. In the event that any or all of these assumptions are subverted (or any cryptographic assumption is found not to hold), the correctness of the final tally can still be ascertained through the independent verification mechanism described in this section.

3.2.5 Protocol

Setup Phase

The trustees in set T generate their threshold trustee keys and initialize the bulletin board $\mathcal{BB}$ using candidate list L, security parameter p, number of ballots to be generated b, valid trustee threshold list τ, code alphabet Σ, code length l, and a heuristic security parameter I where

$\{L, p, b, \Sigma, l, \tau, I\}$ is issued to T by an external entity not considered herein. The audit described in this paper requires the commitment of the voting system to several consistent back-ends, each of which can be used to tally votes from the confirmation numbers. I is the number of back-ends constructed by the system.

Let the notation $X_i(j) = x_{i,j}$ denote the j-th element in the i-th instance of a shuffled list X_i. Additionally let the notation X_i', X_i'' and X_i''' denote list X shuffled by the composition of permutations $(\pi_{(i,1)})$, $(\pi_{(i,2)} \circ \pi_{(i,1)})$, and $(\pi_{(i,3)} \circ \pi_{(i,2)} \circ \pi_{(i,1)})$, respectively.

Using a trusted computing platform, the trustees perform the following computations:

1. Initialize security parameters: $P \leftarrow \mathsf{Parameters}(1^p)$.

2. Initialize bulletin board: Post $\{P, L, p, b, \Sigma, l, \tau, I\}$. and the specification of all functions to $\mathcal{BB}$.

3. Generate trustee keys: Each trustee T_i contributes entropy ω_i and is issued corresponding trustee key k_i via an untappable channel with the trusted computing platform $(k_1, k_2, \ldots, k_t) \leftarrow \mathsf{TrusteeKeys}(\omega_1, \ldots, \omega_t, \tau, \lambda)$.

4. Generate election key: assuming the trusted platform is stateful during this phase, the election master key K is generated by the previous step. (Note that key K must not leave or be leaked from the trusted platform during computation, nor should the trusted platform be stateful between the setup, result declaration, and audit response phases. A minimum of τ keys from $\{k_1, k_2, \ldots, k_t\}$ can regenerate all the information required for the result declaration and audit response phases.)

5. Generate ballots: the trusted platform computes

$$\mathbf{P} = \{\mathbf{Q}, \mathbf{R}, \mathbf{S}\} \leftarrow \mathsf{GenerateBallots}(L, \Sigma, l, K, b)$$

and transmits **P** via a private channel to a trusted printing service that produces paper ballots with corresponding serial numbers and confirmation numbers in revealing ink. Note that initially, the recorded voter marks table **R** is empty.

6. Shuffle P and cryptographically commit to the shuffles:

   The following mixnet-like construction shuffles the two lists **Q** and **S** and posts commitments to the two shuffled lists and to the shuffles. The shuffles are constructed in a manner that will make the tally-verification audit simple to implement, as will be seen later. See Figure 3.4 for an illustration on the example of Figure 3.2. Note that, *in this example only*, we use cyclic permutations and a swap *merely* in an attempt to illustrate the mixnet-like construction in as simple a manner as possible. We *do not* advocate the restriction of permutations to a set of a few permutations, but, as mentioned below, require that each permutation be chosen in a pseudorandom manner from the set of all possible permutations of the respective tables.

   (a) Generate permutations: For each back end, the trusted platform computes three permutations. That is, for $0 \leq i < I$, the trusted platform computes:

   - $\pi_{(i,1)} \leftarrow \mathsf{Perm}(\mathsf{Subkey}(K, \{\text{"1st"}, i\}), (b \cdot n))$
   - $\pi_{(i,2)} \leftarrow \mathsf{Perm}(\mathsf{Subkey}(K, \{\text{"2nd"}, i\}), (b \cdot n))$
   - $\pi_{(i,3)} \leftarrow \mathsf{Perm}(\mathsf{Subkey}(K, \{\text{"3rd"}, i\}), (b \cdot n))$

   (b) Shuffle lists: For each back end the trusted platform computes a single-shuffled instance $\mathbf{Q}'_i$ of **Q** and a triple-shuffled instance $\mathbf{S}'''_i$ of **S**. (Note that the number of apostrophes denotes the number of shuffles the list has gone through). That is, for $0 \leq i < I$ and $0 \leq j < (b \cdot n)$, the trusted platform computes:

   - $\mathbf{Q}'_i(j) \leftarrow \mathbf{Q}(\pi_{(i,1)}(j))$

- $\mathbf{S}_i'''(j) \leftarrow \mathbf{S}_i(\pi_{(i,3)}(\pi_{(i,2)}(\pi_{(i,1)}(j))))$

(c) Commitments: The trusted platform commits to each back end—the shuffled confirmation number numbers, corresponding candidate lists, and permutation values—on an element-by-element basis. For each single-shuffled code list $\mathbf{Q}_i'(j) = \{\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j}, q_{i,j}\}$, triple-shuffled candidate list $\mathbf{S}_i'''(j) = s_{i,j}$, and the corresponding elements of permutations $\pi_{(i,h)}(j)$, the trusted platform computes commitments as follows:

For $1 \leq h \leq 3$, $0 \leq i < I$ and $0 \leq j < (b \cdot n)$,

- $\bar{\alpha}_{i,j} \leftarrow \mathsf{Commit}(\mathsf{Subkey}(K, \{\text{“}\alpha\text{”}, i, j\}), \alpha_{i,j})$
- $\bar{\beta}_{i,j} \leftarrow \mathsf{Commit}(\mathsf{Subkey}(K, \{\text{“}\beta\text{”}, i, j\}), \beta_{i,j})$
- $\bar{\gamma}_{i,j} \leftarrow \mathsf{Commit}(\mathsf{Subkey}(K, \{\text{“}\gamma\text{”}, i, j\}), \gamma_{i,j})$
- $\bar{q}_{i,j} \leftarrow \mathsf{Commit}(\mathsf{Subkey}(K, \{\text{“}q\text{”}, i, j\}), q_{i,j})$
- $\bar{s}_{i,j} \leftarrow \mathsf{Commit}(\mathsf{Subkey}(K, \{\text{“}s\text{”}, i, j\}), s_{i,j})$
- $\bar{\mathbf{Q}}_i'(j) \leftarrow \{\bar{\alpha}_{i,j}, \bar{\beta}_{i,j}, \bar{\gamma}_{i,j}, \bar{q}_{i,j}\}$
- $\bar{\mathbf{S}}_i'''(j) \leftarrow \bar{s}_{i,j}$
- $\bar{\pi}_{(i,h)}(j) \leftarrow \mathsf{Commit}(\mathsf{Subkey}(K, \{\text{“}\pi\text{”}, h, i, j\}),$ $\pi_{(i,h)}(j))$

(d) The trusted platform publishes all $\bar{\mathbf{Q}}_i'$, $\bar{\mathbf{S}}_i'''$, and $\bar{\pi}_{(i,h)}$ to $\mathcal{BB}$.

7. The trusted platform's internal state is purged.

Voting Phase

A voter V, upon being successfully authenticated by election judges, is given a ballot $B_g = \{\alpha_g, \beta_g, \gamma_g, q_{gn}, q_{gn+1} \ldots, q_{gn+n-1}, s_0, s_1, \ldots, s_{n-1}\}$ consisting of a serial number

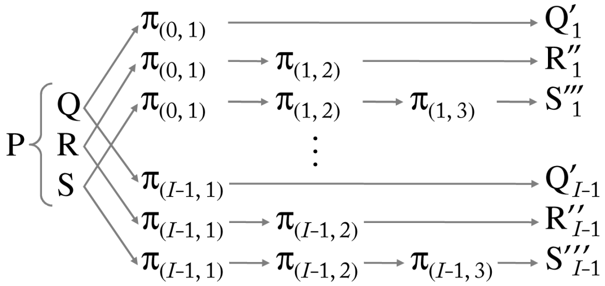


Figure 3.3: Master table **P** links serial numbers (**Q**) with voter marks (**R**) for candidates (**S**). **P** is obfuscated by applying three independent random shuffles $\pi_{(i,1)}, \pi_{(i,2)}$, and $\pi_{(i,3)}$ to each table, as shown above. I independent instances of these shuffled tables are created and cryptographically committed to. For each instance i, either $\pi_{(i,2)}$ or $\pi_{(i,3)}$ can be challenged by the public to be revealed. In this way the correctness of **P** can be audited without directly revealing the links between elements in **Q**, **R**, and **S**.

α_g printed in an optical-scan readable "barcode" and selection/candidate list $s_0, \ldots, s_{n-1}$ printed in normal ink. Serial numbers $\beta_g and \gamma_g$ and the corresponding confirmation numbers $q_{gn}, q_{gn+1}, \ldots, q_{gn+n-1}$ are printed in invisible ink.

To vote, V marks the optical scan bubble beside the desired selection s_d using the decoder pen, which reveals the confirmation number q_{gn+d}.

Upon scanning ballot B_g, the optical scanner produces an "electronic ballot image" $EBI_g = \{\alpha_g, r_{gn}, r_{gn+1}, \ldots, r_{gn+n-1}\}$, where $r_{gn+d} = 1$ if and only if a darkened region (*i.e.,*a mark) was detected inside the optical scan bubble beside the d-th selection s_d. All other $\hat{d} \neq d$ register $r_{gn+\hat{d}} = 0$. The specific electoral system in use would dictate how many marks (*i.e.,*distinct d's) are permissible on a single ballot. V can then choose to construct a vote receipt $VR_g = \{\beta_g, \gamma_g, q_{gn+d}\}$ for each s_d marked.

Instead of voting on a particular ballot, V can select it to be "print audited" in accordance with the procedures specified in Section 3.1. All confirmation numbers are revealed, and **one of** $\{\beta_g, \gamma_g\} = \delta_g$ is revealed. The print-audited ballot becomes $PA_g = \{\delta_g, q_{gn}, q_{gn+1}, \ldots, q_{gn+n-1}, s_0, s_1 \ldots, s_{n-1}\}$. For example, if Ballot 0001 of Figure

j	α	β	γ	q
0	0000	7973	4630	WT9
1	0001	2567	1490	J3K
2	0001	2567	1490	TC3
3	0002	4900	7891	9JH
4	0002	4900	7891	J3K
5	0003	1631	5275	KWK
6	0003	1631	5275	H7T
7	0000	7973	4630	7LH

(a) Table $\mathbf{Q}'_1$

j	s
0	Bob
1	Alice
2	Bob
3	Alice
4	Bob
5	Alice
6	Alice
7	Bob

(b) Table $\mathbf{S}'''_1$

Figure 3.4: Tables $\mathbf{Q}'$ and $\mathbf{S}'''$ for the example of Figure 3.2. $\mathbf{Q}'$ is $\mathbf{Q}$ permuted by $\pi_{(1,1)}$, which is an upward circular shift of one unit, and $\mathbf{S}'''$ is $\mathbf{S}$ permuted by $\pi_{(1,3)} \circ \pi_{(1,2)} \circ \pi_{(1,1)}$, where $\pi_{(1,2)}$ corresponds to an upward circular shift of two units, and $\pi_{(1,3)}$ swaps the last two elements in the list. Note that the confirmation numbers of $\mathbf{Q}'$ can be made to match up with the correct candidates in $\mathbf{S}'''$ if the permutation $\pi_{(1,3)} \circ \pi_{(1,2)}$ is applied to $\mathbf{Q}'$. Note also that we use simple permutations such as these merely for the purposes of illustration. For the system itself, we advocate that each permutation be chosen pseudorandomly from the set of all possible permutations, without restricting this set to the set of simple permutations such as cyclic permutations or swaps.

3.2 were not voted, but were print audited, and β_1 revealed, the print-audited ballot would be $PA_{0001} = \{1490, \texttt{J3K}, \texttt{TC3}, \texttt{Alice}, \texttt{Bob}\}$. (Note that print-audited ballots cannot be voted).

Declaring Results

After the polling concludes, a valid subset of trustees (as defined by τ) assembles to tally and declare the results. The trustees also make available data regrading the tally processing that will be audited in the audit phase. Given the set of all EBIs recorded during the election, the trustees proceed using the trusted platform as follows:

1. Regenerate election master key: Each trustee T_i transmits her trustee key k_i over an untappable channel to the trusted platform. The election master key K is reconstructed

by calling $\mathsf{ElectionKey}(\{j_1, \ldots, j_n\})$ if at least τ trustees supply correct keys (where $\{j_1, j_2, \ldots, j_n\}$ are n keys provided by n trustees).

2. Regenerate ballot list: Ballot list **P** is reconstructed by rerunning step 5) of the setup phase,

3. Construct list of recorded marks: For each EBI_g recorded during the election, populate **R** by setting $\mathbf{R}(gn + j) = r_{gn+j}$ for $0 \leq j < n$ and $r_{gn+j} \in EBI_g$. Any unused, spoiled or print-audited ballot B_g inherently constitutes an EBI_g with all $r_{gn+j} = 0$.

4. Post voted codes: During the dispute resolution period (described in Section 3.1), all voted codes shall be published. For all $\mathbf{R}(gn + d) = 1$ post $\{\beta_{gn+d}, \gamma_{gn+d}, q_{gn+d}\}$ and corresponding commitment keys.

5. Post results: Using **P**, tabulate the election results and post them to $\mathcal{BB}$.

6. Post double-shuffled marks list for audit purposes:

   (a) Regenerate permutations: For $0 \leq i < I$ and $0 \leq j < (b \cdot n)$, the trusted platform recomputes permutations:

   - $\pi_{(i,1)} \leftarrow \mathsf{Perm}(\mathsf{Subkey}(K, \{\text{“1st”}, i\}), (n \cdot b))$
   - $\pi_{(i,2)} \leftarrow \mathsf{Perm}(\mathsf{Subkey}(K, \{\text{“2nd”}, i\}), (n \cdot b))$

   (b) Shuffle Lists: For $0 \leq i < I$ and $0 \leq j < (b \cdot n)$, the trusted platform computes I independent double-shuffled instances $\mathbf{R}''_i$ of $\mathbf{R}_i$:

   - $\mathbf{R}''_i(j) \leftarrow \mathbf{R}(\pi_{(i,2)}(\pi_{(i,1)}(j)))$

   (c) The trusted platform publishes all $\mathbf{R}''_i$ to $\mathcal{BB}$ and purges its internal state.

See Figure 3.5 for an illustration using the example of Figures 3.2 and 3.4.

α	β	γ	q
0000	7973	4630	WT9
0001	2567	1490	TC3
0002	4900	7891	9JH
0003	1631	5275	H7T

(a) Revealed values of Table **Q**

j	r
0	1
1	1
2	0
3	0
4	1
5	0
6	1
7	0

(b) Table $\mathbf{R}_1''$

Figure 3.5: The revealed confirmation numbers, (entries in **Q**) and revealed table $\mathbf{R}_1''$, which is a shuffled version of **R**, using the permutation $\pi_{(1,2)} \circ \pi_{(1,1)}$, where $\pi_{(1,1)}$ is an upward circular shift of one unit, and $\pi_{(1,2)}$ corresponds to an upward circular shift of two units. The tally will be “three votes for `Alice` and one vote for `Bob`.” The permutations used are secret. Note that if $\mathbf{R}_1''$ is permuted by $\pi_{(1,3)}$, the votes will be listed with regard to candidate list $\mathbf{S}'''$ of Figure 3.4. If list $\mathbf{Q}_i'$ of the same figure is permuted by $\pi_{(1,2)}$, the confirmation numbers will be listed as corresponding to the choices of $\mathbf{R}_1''$ above. Note also that we use simple permutations for the purposes of illustration. For the system itself, we advocate that each permutation be chosen pseudorandomly from the set of all possible permutations, without restricting this set to the set of simple permutations such as cyclic permutations or swaps.

Note that, at this stage, the list $\mathbf{R}''_i$ is such that, if permuted by $\pi_{(i,3)}$, the votes will be listed as obtained for the candidate list $\mathbf{S}'''_i$. Further, if list $\mathbf{Q}'_i$ is permuted by $\pi_{(i,2)}$, the confirmation numbers will be listed in the order of the votes $\mathbf{R}''_i$.

Audit Challenge and Response

In order to ensure robust, correct behavior by the trustees and in turn, the correctness of the election outcome, two audits are carried out. We first describe the tally computation audit. For each back-end committed to by the trustee, a coin flip determines whether the trustees will demonstrate that the ballot marks of the corresponding public table $\mathbf{R}''_i$ correspond correctly to (a) the announced tally or (b) the public confirmation numbers for voted ballots. This is done by opening the commitments to the permutation $\pi_{(i,3)}(j)\ \forall\ j$ or to the permutation $\pi_{(i,2)}(j)\ \forall\ j$ respectively. Second, we describe the print audit. For values of j, in the original ballot list $\mathbf{Q}$, corresponding to print-audited ballots, permutation values $\pi_{(i,2)}(j)$ and $\pi_{(i,3)}(j)$ are opened $\forall i$. We now describe these audits in more detail.

1. Public challenge of trustees: some time after the trustees have completed declaring the results and posting the shuffled marks lists, each instance of $\mathbf{Q}'_i, \mathbf{R}''_i, \mathbf{S}'''_i$ is challenged to be partially revealed for the purposes of auditing. A fair public coin $\mathcal{C}$ is tossed I times providing a series of audit challenges $\mathcal{C} \in \{0,1\}^I$, which are posted to $\mathcal{BB}$.

2. For the tally computation audit: For $0 \leq i < I$ and $0 \leq j < (b \cdot n)$, the trusted platform performs the following actions:

   (a) If $\mathcal{C}(i) = 0$, regenerate and publish the confirmation numbers $\mathbf{Q}'_i$ and the association between $\mathbf{Q}'_i$ and $\mathbf{R}''_i$. That is, regenerate and publish the following:

   - The second permutation $\pi_{(i,2)}$

- The commitment subkeys of

  $\pi_{(i,2)} : \kappa_{\pi_{(i,2)}}(j) \leftarrow \mathsf{Subkey}(K, \{\text{“}\pi\text{”}, 2, i, j\})\ \ \forall j$

- The commitment subkeys to all elements of

  $\mathbf{Q}'_i : \kappa_{x_{i,j}} \leftarrow \mathsf{Subkey}(K, \{\text{“x”}, i, j\})$ where x $= \{\alpha, \beta, \gamma, q\}\ \ \forall j$

(b) If $\mathcal{C}(i) = 1$, regenerate and publish the permuted candidate list $\mathbf{S}'''_i$ as well as the association between $\mathbf{R}''_i$ and $\mathbf{S}'''_i$. That is, regenerate and publish the following:

- The third permutation $\pi_{(i,3)}$
- The commitment subkeys of

  $\pi_{(i,3)} : \kappa_{\pi_{(i,3)}}(j) \leftarrow \mathsf{Subkey}(K, \{\text{“}\pi\text{”}, 3, i, j\})\ \ \forall j$

- The commitment subkeys to all elements of

  $\mathbf{S}'''_i : \kappa_{s_{i,j}} \leftarrow \mathsf{Subkey}(K, \{\text{“s”}, i, j\})\ \ \forall j$

3. For the ballot audit, compute all permutation elements and commitment keys not computed in the tally audit and required for the purposes of demonstrating the entire path of the ballot through the mixnet-like construction. That is, the trusted computing platform does the following for $0 \leq i < I$:

(a) If $\mathcal{C}(i) = 0$:

- Regenerate $\pi_{(i,3)}$ (do not publish it)
- For each ballot $PA_g = \{\delta_g, q_{gn}, q_{gn+1}, \ldots, q_{gn+n-1}, s_0, s_1 \ldots, s_{n-1}\}$ that is print audited:

  i. Search for all elements in $\mathbf{Q}'_i$ such that the second component is δ_g. If there is no such element, search for all elements in $\mathbf{Q}'_i$ such that the third component is δ_g. That is, find all j' such that $\mathbf{Q}'_i(j') = \{*, \delta_g, *, *\}$, failing which find all j' such that $\mathbf{Q}'_i(j') = \{*, *, \delta_g, *\}$. For all such j':

- Compute $j''' \leftarrow \pi(i,3)(j'')$ where $j'' \leftarrow \pi(i,2)(j')$ has already been computed in the tally computation audit.
- Publish $\pi(i,3)(j'')$. Compute and publish the subkey used to commit to $\pi_{(i,3)}(j'')$: $\kappa_{\pi_{(i,3)}}(j'') \leftarrow \mathsf{Subkey}(K, \{``\pi", 3, i, j''\})$.
- Publish $\mathbf{S}_i'''(j''')$: Compute and publish the commitment subkey for this value: $\kappa_{s_{i,j'''}} \leftarrow \mathsf{Subkey}(K, \{``\text{s}", i, j'''\})$.

(b) If $\mathcal{C}(i) = 1$:

- Regenerate $\pi_{(i,2)}$ (do not publish it).
- For each ballot $PA_g = \{\delta_g, q_{gn}, q_{gn+1}, \ldots, q_{gn+n-1}, s_0, s_1 \ldots, s_{n-1}\}$ that is print audited:

  i. Search for all elements in $\mathbf{Q}_i'$ such that the second component is δ_g. If there is no such element, search for all elements in $\mathbf{Q}_i'$ such that the third component is δ_g. That is, find all j' such that $\mathbf{Q}_i'(j') = \{*, \delta_g, *, *\}$, failing which, find all j' such that $\mathbf{Q}_i'(j') = \{*, *, \delta_g, *\}$. For all such j':

  - Compute $j'' \leftarrow \pi(i,2)(j')$. Note that $j''' \leftarrow \pi(i,3)(j'')$ has already been computed in the tally computation audit.
  - Publish $\pi_{(i,2)}(j')$. Compute and publish the subkey used to commit to $\pi_{(i,2)}(j')$: $\kappa_{\pi_{(i,2)}}(j') \leftarrow \mathsf{Subkey}(K, \{``\pi", 2, i, j'\})$.
  - Publish $\mathbf{Q}_i'(j')$. Compute and publish the commitment subkey for this value: $\kappa_{x_{i,j'}} \leftarrow \mathsf{Subkey}(K, \{``\text{x}", i, j'\})$ where x $= \{\alpha_g, \beta_g, \gamma_g, q\}$.

3.2.6 Correctness Proofs

We summarize the proofs of correctness that verifying agents A can perform and explicitly state conditions under which the proof completes successfully. In general, the best practice response to proofs that do not complete successfully (*i.e.,*fail) is an open policy question and is not considered here. In case of voter receipts, however, a failed receipt check has a dispute resolution process described in section 3.1.3.

Note that, for the print audit and tally check correctness proofs, A will verify commitments. In particular, A will confirm that all commitment keys that were challenged as a result of the challenge coin tosses and the print audit were responded to (*i.e.,*published on $\mathcal{BB}$) during steps 2 and 3 in the previous section. For all commitment keys κ_x to message x posted to $\mathcal{BB}$ during the audit, A searches $\mathcal{BB}$ for the corresponding message x and commitment value $\bar{x}$ and tests whether $\mathsf{Decommit}(\kappa_x, x, \bar{x})$ outputs 1 (valid). This verification step is successful if and only if all of A's executions of $\mathsf{Decommit}()$ output 1.

Receipt Check

For all challenges $\mathcal{C}(i) = 0$ and $0 \leq j < (b \cdot n)$, A locates permutations $\pi_{(i,2)}$, code lists $\mathbf{Q}'_i$, and recorded mark lists $\mathbf{R}''_i$ on $\mathcal{BB}$. A reconstructs the assertion of the voting system, that $\mathbf{Q}'_i(j)$ is marked or not marked as indicated by the mark value $\mathbf{R}''_i(\pi_{(i,2)}(j))$.

This verification step successfully verifies voter receipt $VR_g = \{\beta_g, \gamma_g, q_{gn+d}\}$ if and only if A is able to conclude that all reconstructed assertions agree with VR_g. Specifically, $0 \leq i < I$, VR_g is said to agree with the assertions if $\{\beta_g, \gamma_g, q_{gn+d}\}$ exists at position j in $\mathbf{Q}'_i$, if $\mathbf{R}''_i(\pi_{(i,2)}(j)) = 1$, and if all other occurrences of β_g and γ_g (that is, all $n-1$ other values of tuples $\{\beta_g, \gamma_g, q_{gn+\hat{d}}\}$ found at positions $\mathbf{Q}'_i(\hat{j})$ correspondingly show recorded mark $\mathbf{R}''_i(\pi_{(i,2)}(\hat{j})) = 0$.

Print Audit

A reconstructs assertions of the code-candidate associations of each print-audited ballot. For all i, $0 \leq i \leq I$, and for each print-audited ballot:

$$PA_g = \{\delta_g, q_{gn}, q_{gn+1}, \ldots, q_{gn+n-1}, s_0, s_1 \ldots, s_{n-1}\}$$

A uses the following procedure:

1. A searches for all elements in $\mathbf{Q}'_i$ such that the second component is δ_g. If there is no such element, A searches for all elements in $\mathbf{Q}'_i$ such that the third component is δ_g. That is, A finds all j' such that $\mathbf{Q}'_i(j') = \{*, \delta_g, *, *\}$, failing which, A finds all j' such that $\mathbf{Q}'_i(j') = \{*, *, \delta_g, *\}$. For all such j':
    - A locates permutation element $\pi_{(i,2)}(j')$, computes $j'' \leftarrow \pi_{(i,2)}(j')$, locates permutation element $\pi_{(i,3)}(j'')$, computes $j''' \leftarrow \pi_{(i,3)}(j'')$.
    - A locates $\mathbf{R}''_i(j'')$ and $\mathbf{S}'''_i(j''')$.
    - The assertion is that $\mathbf{Q}'_i(j')$ is the confirmation number corresponding to the candidate $\mathbf{S}'''_i(j''')$ and that the unique ballot with one serial number δ_g has not been voted.
2. This verification step successfully verifies the print-audited ballot if it agrees with the assertions. That is, if A is able to obtain n values of j' and conclude that, $\forall j'$:
    - The corresponding commitments were opened correctly,
    - $\{\delta_g, q_{gn+\iota}\} \in \mathbf{Q}'_i(j')$ for some ι such that $0 \leq \iota \leq n-1$ and that each value of ι corresponds to exactly one value of j'.
    - $\mathbf{R}''_i(j'') = 0$.

- $\mathbf{S}_i'''(j''') = s_\iota$.

Tally Check

1. A will check that the corresponding commitments were opened correctly.

2. A will verify $\mathcal{BB}$ self-consistency:

   (a) For all challenges $\mathcal{C}(i) = 0$ and $0 \leq j < (b \cdot n)$, A locates permutations $\pi_{(i,2)}$, code lists $\mathbf{Q}_i'$, and recorded mark lists $\mathbf{R}_i''$ on $\mathcal{BB}$. A reconstructs the assertion of the voting system that $\mathbf{Q}_i'(j)$ is marked or not marked as indicated by the mark value $\mathbf{R}_i''(\pi_{(i,2)}(j))$. This verification step successfully verifies $\mathcal{BB}$ self-consistency if all public voted confirmation numbers correspond exactly to $\mathbf{R}_i''(\pi_{(i,2)}(j)) = 1$.

   (b) For all $\mathcal{C}(i) = 1$, A locates $\pi_{(i,3)}$, $\mathbf{R}_i''$ and $\mathbf{S}_i'''$ on $\mathcal{BB}$. For all j, A reconstructs the assertion of recorded mark $\mathbf{R}_i''(j)$ made for candidate $\mathbf{S}_i'''(\pi_{(i,3)}(j))$ and computes the election outcome by tallying each of these assertions. He checks the declared tally against the computed tally. This verification step successfully verifies $\mathcal{BB}$ self-consistency if the two tallies are identical.

3.3 Security Analysis

In the previous section, we described the verification proofs for receipt checks, the tally check, and print-audited ballots. In this section, we both quantify the effectiveness of the verification and consider the security of Scantegrity from additional attacks, most involving a procedural element not easily captured by a cryptographic description. Thus, the goal of this analysis is to sketch the security heuristics underlying the design rather than to rigorously prove security properties in a formal cryptographic model.

We consider three categories of attacks. The first category is *manipulation attacks*, in which the goal of the attacker is to manipulate the final tally so that the election's outcome is more favorable to the attacker's preferred candidate(s). The second is *identification attacks*, where the goal of the attacker is to form a link between voting intent and ballot receipts. The final category is *disruption attacks*, in which the attacker wishes to prevent the completion or certification of the election. Since disruption attacks are applicable to any voting system and difficult to prevent, we will consider only a special case of disruption involving the prevention of certification of any tally in the event the attacker feels the results may be unfavorable.

As an enhancement to optical scan, Scantegrity is inherently constrained by our design goal of non-interference with the underlying optical scan processes. For this reason, Scantegrity is designed to be a strict improvement over optical scan systems with manual recounts. However, components that cannot be secured without intervening in the underlying processes of optical scan are not pursued.

3.3.1 Assumptions

The level of security of Scantegrity depends on the nature of the attack. Critical components offer probabilistic security that is invariant to the adversary's computational power, while other components premise their security on one or more assumptions, both procedural and cryptographic in nature. The security setting of our analysis includes the following assumptions,

1. A trusted computing platform exists for use by election officials (*contra* identification attacks).

2. The set of collusive officials in the election authority does not satisfy the threshold requirement for recovery of the master key (identification and disruption).

3. Chain-of-custody over the printed ballots prior to voting day (identification).[3]

4. The inability of voters and others to read codes printed in invisible ink (manipulation, identification).

5. The pollbook is properly balanced (manipulation),

6. The intractability of obtaining information about a message given only its cryptographic commitment (identification), and

7. The intractability of opening a cryptographic commitment of a message differs from the message initially committed (manipulation).

In our view, most of these assumptions are reasonable and standard in the literature. The trusted platform is a scaled-down computing device, with no external memory, running software attested by the trustees that performs the cryptographic operations. Trustees could be selected from competing political parties to avoid collusion. Using a threshold scheme allows the election to proceed even if a group of trustees is unable, or refuses, to supply their key share. Prior to the election, printed ballots must be protected against an adversary revealing codes and reprinting substitute ballots. Assumption 4 is unique to our approach and we provide justification for it in Section 3.4.1. “Balancing the pollbook” refers to the

[3] Note that this assumption is substantially weaker than requiring chain of custody throughout the entire election. We only require ballots be stored and delivered to the polling place, which does not require they be handled individually.

assumption that the sum of the number of voted, tallied, and spoiled ballots is equal to the number of cast ballots, which is not larger than the number of voters. Assumptions 6 and 7 are referred to as the hiding and binding properties of commitments, respectively, in the previous section.

3.3.2 Manipulation Attacks

Manipulation attacks are those in which an adversary attempts to manipulate final results. This can be accomplished by misprinting ballots, coercing voters, or misreporting the published records.

Printing

An adversary may misprint ballot B_g so that the code $q_{gn+\hat{d}}$ associated with candidate $s_{\hat{d}}$ in the master list **P** is printed beside a different candidate s_d (or all candidates) on the same ballot. If the adversary then modifies any EBI_g associated with such a misprinted ballot such that $r_{gn+\hat{d}} = 1$ and $r_{gn+d} = 0$, the system will count the vote for $s_{\hat{d}}$ and report q_{gn+d} as the confirmation number, which is consistent with what appears on the ballot.

The print audit mechanism, described in section 3.2, is designed to make such an attack detectable by revealing discrepancies between printed ballots and $\mathbf{Q}'_i$ using commitments $\bar{\mathbf{Q}}'_i$ under assumption 7. If the number of ballots chosen to be print-audited is $0 \leq b_a \leq b$ where b is the number of ballots in the election overall, the probability of detecting at least 1 of $1 \leq b_f \leq b$ misprinted ballots is:

$$
\begin{aligned}
\mathbf{Pr}[\text{detection}] &= 1 - \frac{\binom{b-b_f}{b_a}}{\binom{b}{b_a}} \\
&= \frac{(b-b_f)!(b-b_a)!}{b!(b-b_f-b_a)!} \qquad (3.9)
\end{aligned}
$$

Voting

One line of manipulation attack can exist in systems that are not diligent in spoiling ballots [88, 12]. If an attacker has a line of communication with the voter, the voter can be instructed to mark her ballot and wait for further instruction. The attacker then instructs the voter to either spoil the ballot or cast it. If the spoiled ballot is not protected or destroyed, the attacker may consult it to see how the voter would have voted had the attacker instructed the voter to cast the ballot. The line of communication can be eliminated by using random material on the ballots to determine the instruction, in a way analogous to the approach of making interactive protocols non-interactive. Scantegrity avoids this line of attack by having spoiled ballots shredded in front of the voter, without the election judge seeing the contents of the ballot.

A second line of manipulation attack can exploit the presence of undervoted ballots. An attacker may add additional marks to a contest left empty by the voter during a recount or appropriately modify the digital records.[4] This attack is not introduced by Scantegrity and exists in any optical scan voting system. One method of prevention is to require each voter to mark a “none of the above” selection when denoting an undervote. Similarly, an attacker might try to prevent a correctly-cast ballot from being tallied by overvoting it; this attack is prevented by not allowing any overvoted ballots to be cast.

[4] Although this issue was previously known to the authors, we acknowledge David Wagner for raising it in private correspondence.

Auditing

Consider a manipulation attack based on swapping voter-made marks in $\mathbf{R}''_i$ from one candidate to the attacker's preferred candidate. To prevent this attack, with probability $1/2$, each back end will be challenged to open the correspondence between the lists $\mathbf{R}''_i$ and $\mathbf{Q}'_i$; any modified mark states for these instances will be incongruent with the voter receipts. The attacker may gamble, only modifying marks in roughly half of the back-end instances in the hope that exactly these will have challenge $\mathcal{C}(i) = 1$ and thus that, instead, that the correspondence between the lists $\mathbf{R}''_i$ and $\mathbf{S}'''_i$ is instead revealed in the modified instances i. The probability of doing so is 2^{-I}. However if a different subset is revealed, the tallies across the subsets will differ and the attack is detectable. Alternatively, the attacker might modify $\mathbf{R}''_i$ for all instances $1 \leq i \leq I$, which guarantees self-consistent tallies but also guarantees the attack is detectable by the receipt check protocol. At first glance this may seem to be an irrational strategy until one considers the possibility of only a small subset of voters actually checking their receipts. With I instances, $b_r \leq b$ ballots actually cast, b_c ballot receipts checked, and b_m modifications to each $\mathbf{R}''_i$, the probability of detection is $(b_r - b_m)!(b_r - b_c)!/b_r!(b_r - b_m - b_c)!$. The adversary will choose the least detectable of the two strategies, thus,

$$\mathbf{Pr}[\text{det.}] \quad = \quad \min(1 - \frac{1}{2^I}, 1 - \frac{(b_r - b_m)!(b_r - b_c)!}{b_r!(b_r - b_m - b_c)!}). \tag{3.10}$$

By estimating b_c and bounding b_m as half of the smallest margin of victory we can certify an election for which we can use this equation to determine a suitable I for our implementation such that the first term exceeds the estimated value of the second. In most instances, $I = 10$ is suitable.

A second approach to manipulating the tally is to change the final state of the ballots. Ballots can have one of three states: voted, print-audited, or spoiled. Under assumption 5, we assume that modifications must preserve the number of ballots in each state. If a voted ballot is maliciously modified to be spoiled, a spoiled ballot must be converted into a voted ballot. To prevent these transitions, the voter retains positive evidence of ballots being in a voted state: knowledge of both serial numbers, $\{\beta_g, \gamma_g\}$. Alternatively, for a print-audited ballot, the voter retains positive evidence a ballot was print-audited via knowledge of all the confirmation numbers on the ballot, $\{q_{gn}, q_{gn+1} \dots q_{gn+n-1}\}$, but only *one of* $\{\beta_g, \gamma_g\}$. Both pieces of information would be unknown to the voter if the ballot were in any other state when the voter left the polling place.

In the case of spoiled ballots, the voter does not retain anything. However, if a spoiled ballot is maliciously converted into a voted ballot, a voted ballot will need to be spoiled, and the corresponding voter can prove malfeasance through knowledge of both chit serial numbers.

The transition from a spoiled to print-audited state is important for different reasons. This transition does not change the tally directly, however it is indirectly useful in facilitating the second manipulation attack presented in Section 3.3.2. By misreporting a spoiled ballot as print-audited, the confirmation codes on the ballot would be released during the verification process allowing a coercer to see if the ballot matched the conditions of the contract for spoiling the ballot. Under assumption 5, this attack will be detectable as it requires a print-audited ballot to be made into a spoiled ballot. To prevent this attack, the trustees could first publish a list of ostensibly spoiled ballots prior to releasing the print audit confirmation codes. If an auditor discovers her print-audited ballot is in the wrong state, the discrepancy can be caught prior to releasing the confirmation numbers.

3.3.3 Identification Attacks

The earliest opportunity for identification occurs during the election initialization process. Others arrive after ballots are printed or after the election has concluded.

Initialization

Successfully changing or introducing faults into the initialization protocol could generate a permutation of **P** or subsequent lists that is known to the attacker. This is not possible if the protocol is run on a trusted computing platform and assumption 1 holds. Without direct interference with the protocol, the attacker may provide structured data instead of randomness in the protocol. However under assumption 2 and the construction of the threshold key generation scheme, any amount less than the minimum threshold of shares leaks negligible information for the purposes of determining the key.

Printing

After the ballots are printed, a number of identification attacks may be conducted, including the addition of revealing marks on the ballots or revealing the codes on the ballots, recording these codes, and reprinting the ballots with unrevealed ink. The prevention of these attacks is based on assumption 3.

Auditing

After the election has concluded, the data generated and published for voter-verification of the tally must meet the requisite ballot secrecy. Given no information other than the tally, a certain level of information can be obtained about which candidate a voter selected. The tally provides a probability distribution for the possible selections and may even exclude selections, based, for example, on a candidate receiving zero votes. This level of information is often legally required and thus acceptable. If the attacker is provided, in addition, with the information on each voter's receipt, further information is revealed: how many marks the voter made and the codes associated with these marks. Our assertion of ballot secrecy is that no additional information is leaked about the association between a mark and code on a receipt and any element in the set of selections in the tally.

Opening only one of the commitments to either (a) the correspondence between confirmation numbers $\mathbf{Q}'_i$ and voter marks $\mathbf{R}''_i$ or (b) the correspondence between voter marks $\mathbf{R}''_i$ and candidates $\mathbf{S}'''_i$ reveals negligible information about permutations $\pi_{(i,3)}$ or $\pi_{(i,2)}$, respectively, hence the association between $\mathbf{Q}$ and $\mathbf{S}$ is always hidden by one cryptographic permutation. The commitment to the permutation key, if binding, uniquely identifies the permutation; however, reversing the commitment is assumed intractable by assumption 6.

3.3.4 Disruption Attacks

In general, disruption attacks are easy to detect but difficult to prevent. Many of the manipulation attacks could be reconstructed as disruption attacks, and the same mechanisms would detect them. However, as stated, we limit our consideration to disruption for the purpose of preventing the certification of an undesired tally (or an expected undesired tally, e.g., if the information is based on exit polls).

Initialization

During the initialization phase, each trustee in the election authority supplies entropy to seed the random number generator used to generate all the permutation keys and commitment secrets needed in the election. Instead of maintaining state, since the state information would need to remain private, when the tally and audit challenge/response phases are entered, the trustees re-enter their key shares to recreate all the necessary data. To prevent a malicious trustee from withholding his entropy or supplying the wrong entropy, we use a threshold key generation scheme (optionally with robustness to a finite number of errors). Under assumption 2, a suitable threshold will allow the reconstruction of the data despite malicious trustees.

Auditing

During the auditing phase, an attacker may file a spurious dispute about the results of a receipt-check. Since the election authority has committed to the confirmation numbers that appeared on the ballot, it can rule out any claimed codes that did appear on the ballot. Thus, filing a spurious but plausible dispute reduces to randomly guessing another code on the ballot. The election authority can quantify the probability of this and create an appropriate statistical trigger that predicts actual receipt-check problems. Let n be the number of candidates on a candidate list L for a particular race and let Σ^l be the cardinality of the set of unique confirmation numbers. The probability of guessing a plausible code on a voted ballot is $p = (n-1)/(\Sigma^l - 1)$. If D disputes are filed and G are considered plausible, the expected value of G if disputes are fabricated is $\mu = D \cdot p$. We set the trigger value 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 [49]. For any $r > \mu$, $\Pr[G - \mu \geq r] \leq (\mu e/r)^r$.

For example, for 5 candidates, 8,000 possible codes, and 1,000 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.

3.4 Implementation Details

The implementation does not implement all the features of the described system. Specifically, there are three differences from the description:

1. The ink chemistry has different properties. The foreground ink used to create the confirmation number is non-reactive and not slow-reactive. While it is technically possible to have slow-reactive we do not have chemistry expertise and were unable to implement this feature. The lack of this feature has an impact on the privacy of ballots stored in the ballot box. An adversary can look at the printed ballot store and collect voter receipts to determine voter selections.[5]

2. The back end used is the Punchscan back end. However, this back end is functionally equivalent to the one given in the description. We chose not to implement the proposed back end because the Punchscan back end has already been subject to extensive testing both by white box testing and in real election scenarios.

3. There is no accessible interface. While our design expects disabled voters and "builds accessibility in," the actual components that make the system accessible are orthogonal to the other parts of the system. There is additional ink and printer technology involved here, and we were unable to implement these features.

In each of the above cases, the intent is eventually to build or implement the features as described. However, our resources in building these systems are limited, so we focused

[5] Note that this is still a reasonably difficult and time-consuming attack, although it is not as hard as being required to use forensic analysis equipment to determine confirmation numbers.

on building the parts that we could build and building the essentials necessary to test the system in real elections. In all three cases, the deficiency does not affect the experience of most voters.

3.4.1 Invisible Ink Library

In this section, we describe the main categories of threats that might take advantage of the properties of invisible ink, our assumptions about ink properties, and the procedures for printing the inks on the ballot. Greater detail is available in [27].

Threats

Note that the only threats to Scantegrity that take advantage of the limitations of the ink are those that are based on

1. Distinguishing between confirmation numbers and their backgrounds.

   The ability to distinguish would allow:

   (a) voters to claim election fraud falsely, and

   (b) anyone with access to ballots to violate ballot secrecy by connecting confirmation numbers to selections

2. Distinguishing between chit serial numbers and backgrounds.

   The ability to distinguish would allow:

   (a) voters to claim that an uncast ballot was cast, and

   (b) anyone with access to uncast ballots to connect chit serial numbers and confirmation numbers with voter selections (in combination with (1))

3. Distinguishing between the two-dimensional barcode and background.

   The ability to distinguish would allow anyone with access to marked ballots to connect two-dimensional bar-codes with voter selections

Assumptions

The main security assumption about the inks is that the slow and fast-reacting inks used for printing confirmation numbers and oval backgrounds respectively are not distinguishable before, and sufficiently after, they have been marked with the ballot-marking pen ("sufficiently after" is taken to mean that the time period is long enough to allow the slow-reacting ink to react). We make a similar assumption about the indistinguishability of the chit serial numbers and the two-dimensional barcode from their background.

The assumptions we make are about physical properties of chemicals and the detectability of differences. Most chemicals (if not all) can be distinguished from one another through a sufficiently sophisticated test; our arguments are that, for all practical purposes, our assumption holds, and we describe here our efforts to make it more difficult to distinguish among the inks, particularly by the naked or microscopically-aided human eye.

Finally, the ability to distinguish enables voters to make false charges of election fraud and anyone to connect information about ballot choices with confirmation numbers and serial numbers. If voters are assumed not to have access to ballots outside the polling booths or to specialized equipment (including the decoding pen) inside the booths, the indistinguishability assumption is only required to hold with respect to the human eye in order to prevent false charges of election fraud.

Procedures used for printing with the inks

In this section we describe ways in which the indistinguishability assumptions may be defeated and our efforts to preserve indistinguishability. Note that the inks proposed for printing on ballots can be used in regular ink-jet printers.

To prevent the soaking of paper

Any type of ink used by inkjet printers soaks into the paper. Even if the ink used to print the codes would be completely invisible, the soaked paper would allow the codes to be easily read. To avoid this, we use two types of ink: a reacting ink used to print the background of the oval and a slow-reactive ink used to print the confirmation numbers. Both inks have the same color (a light yellow) if printed on the same piece of paper. The reacting ink turns black immediately when it interacts with the ink of the marking pen, while the ink used for the codes undergoes the same reaction at a slower pace. Thus, the immediate result is a yellow confirmation number inside a black oval—the highest-contrast color combination. After several minutes, the slow-reacting ink will have reacted leaving the oval completely black.

To avoid the overlapping of inks

We divide the oval in small square tiles called texels. Each texel is entirely printed with either reactive or slow-reactive ink, but never with a combination of them. A small constant-sized gap is left between any two adjacent texels, such that when two adjacent tiles are printed with different inks, the two inks never overlap even if they diffuse outward as they absorb into the paper. Without such a gap, a border of overlapping types of ink could emerge, under a microscope, for example, making the border easier to detect. Additionally, we ensure that the position of the code in the oval is not fixed; the codes can be shifted left or right.

The addition of confusing fluorescence

The use of special types of radiation can expose invisible inks. We apply a third type of ink that we call a masking ink. It is colorless but has high fluorescence. Masking ink is the last ink sprayed onto the paper. We add random amounts of masking ink to all texels of the oval. This is designed to mask the eventual difference in fluorescence between the reactive ink and slow reactive ink used for the codes, as well as a cover to prevent lifting particles from the paper with tape.

Ballot-marking pens

We conclude this section with some additional importatnt non-security properties of our inks. The ballot-marking pens that we use to mark the ovals have a tip that is wider than the height of the oval. A voter can mark the entire oval using a single strike of the pen which is faster than penciling in the mark. Even if the voter pens in more than the oval, the result is a clean, perfectly filled oval. The use of invisible ink also deters stray dark marks that can confuse scanners, although the light yellow hue of the ink could still be visible. The portion of the chit reserved for the voter to record the confirmation numbers can also have a solid layer of the same reacting ink so that the voter may record the codes with the same pen.

Chapter 4

Mock Election at Takoma Park

On April 11, 2009, ninety-five voters cast ballots on the Scantegrity II voting system during a mock election held at the Community Center in Takoma Park, Maryland, coinciding with Takoma Park's celebration of Arbor Day. The purpose of this exercise was to demonstrate and tune Scantegrity's capability in preparation for the Takoma Park municipal election in November 2009. This chapter, published as [135, 136], describes our experiences using Scantegrity during the Mock election and presents and interprets data collected through questionnaires, unobtrusive observations, and independently-administered focus groups.

There is significant debate in the voting systems community about how easily E2E verifiable election systems can be understood, used, and administered in real elections. However, there is little evidence from which to draw any conclusions. We created Scantegrity to show that E2E systems can be built that are viable to use in modern elections. For our next step, we set out to measure how easy Scantegrity is for voters to use and poll workers to administer, and to study how well voters and poll workers accept this system.

After our initial Scantegrity II proposal [38], Sherman asked the Maryland Board of Elections if there were a municipal government willing to try experimental election systems, and Takoma Park was suggested to our group. He made an initial contact, and later our research team was reached by Takoma Park officials through FairVote.[1]

Takoma Park wanted our research team to show that our system would work for them, so we designed a Mock election to determine the viability of Scantegrity for use in Takoma Park's municipal election. We wanted to measure how easy it is for voters to use and poll workers to administer, and if both groups would accept the system.

Our hypothesis was: *Voters and election officials will accept and have confidence in Scantegrity as a viable, practical high-integrity voting system. They will find it reasonably easy to use and administer compared to traditional optical scan voting. A statistically significant number of voters will verify their votes online, and a statistically significant number of them will detect errors, if present, to produce high assurance in the election outcome.*

Takoma Park accepted our proposal, but added a requirement to mimic a binding governmental election. To meet that requirement we also produced realistic voting procedures to test, which were modified and used in the real election. Unfortunately, such requirements necessarily constrained our research methodologies, but these constraints were necessary to assess the performance of the system and its viability in a binding election.

Would Scantegrity's E2E mechanism be viewed negatively or significantly affect the voting process? What issues will voters have when using the system? Will election judges and voters value the extra security provided by the system? Will enough voters use the E2E receipt? Will voters accept this voting system during the real election?

To answer these questions at the Mock election, we measured Scantegrity's performance through surveys, observations, and focus groups. Eighty voters and all six Takoma Park poll workers filled out questionnaires about their experiences with Scantegrity, including

[1] See: http://fairvote.org

questions about how easy the system was to use and administer and how well they understood and accepted the system. Two unobtrusive observers watched and timed fifty-three of the voters as they voted. A professional moderator led two focus groups: one for all six poll workers and one attended by four voters.

This mock election study provided evidence that the E2E mechanism in Scantegrity would not have significant impacts on the election. It also identified a number of practical issues with our implementation and provided insight into how we could improve the E2E mechanisms and procedures used during the election.

In the next section, we present our methodology in conducting this viability study. In Section 4.2, we present our results from the data collected, and in Section 4.3 we discuss the results. We provide a list of recommended implementation changes in Section 4.4.

4.1 Methods

We now describe the voting and research procedures used in the Mock election. It is important to understand that this study is not a usability study, and it does not share the same goals normally found in such studies. Our dominant interest was in determining if the system would work in a real environment, or what we must do to make the system work in such an envronment.

Our research protocols and questionnaires were approved by UMBC's Institutional Review Board, as required for experiments with human subjects. Polls were open from 10am to 2pm during the mock election.

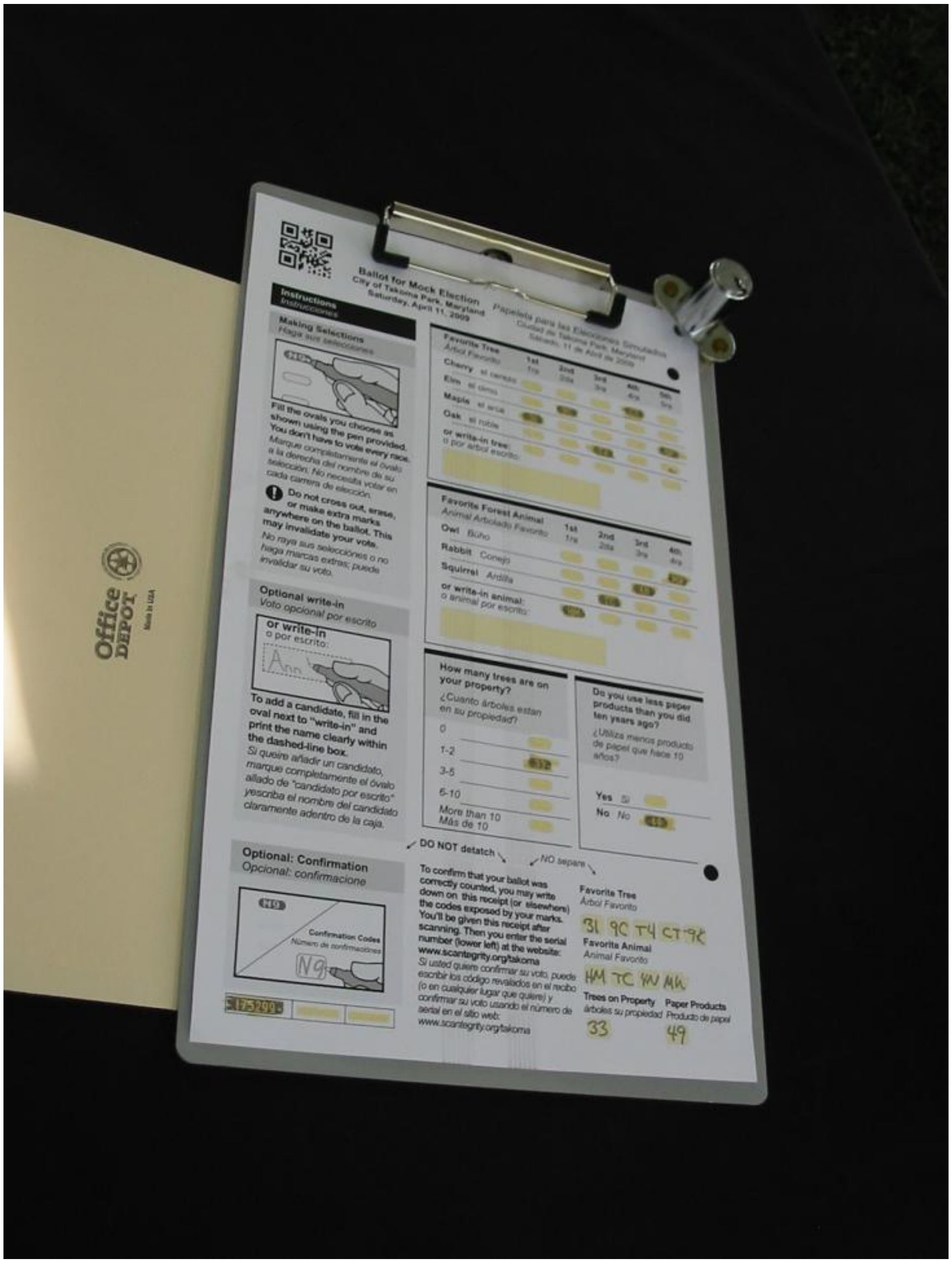
4.1.1 Voter Experience

Each voter first approached a welcome table located outside the polling room. After signing a consent form, the voter proceeded to an adjacent check-in table. There, a poll worker looked up the voter's name in a poll book and issued the voter authority card. The voter then entered the polling room and presented the voting authority card to poll workers at the ballot issue table, who issued a Scantegrity ballot secured to a locked clipboard with privacy sleeve (see Figure 4.1).

The voter proceeded to one of three voting areas, each with a cardboard privacy shield. Using a special pen with revealing ink, the voter marked her ballot choices by marking the selected ovals with the pen. The revealing ink exposed a two-character confirmation number in each marked oval. Optionally, using the special pen, the voter could write down these confirmation numbers on a detachable ballot chit, treated with reactive ink. As required by Takoma Park for municipal elections, Instant Runoff Voting (IRV) [122] was used, so each voter was asked to rank each candidate in order of preference.

Figure 4.2 shows the ballot, which featured four questions about trees. To avoid possible confusion, Takoma Park officials required that races on our ballot not resemble those on official ballots. November's official ballot had two races (mayor and ward council member) per ward. The municipal election can also have ballot questions. So we chose to include two contests and 2 ballot questions on the ballot.

Instead of voting on the issued ballot, each voter had the option of performing a "print audit" to verify that the ballot had been printed correctly. To do so, the voter walked to a voter assistance table and followed instructions from a poll worker. The poll worker marked the ballot spoiled and exposed all confirmation numbers. The voter was permitted to copy information from the ballot to take home. A poll worker then escorted the voter back to the ballot issue table to receive another ballot. Each voter was allowed to receive up to three



Ballot for Mock Election
City of Takoma Park, Maryland
Saturday, April 11, 2009

Papeleta para las Elecciones Simuladas
Ciudad de Takoma Park, Maryland
Sábado, 11 de Abril de 2009

Instructions
Instrucciones

Making Selections
Haga sus selecciones

Fill the ovals you choose as shown using the pen provided. You don't have to vote every race.
Marque completamente el óvalo a la derecha del nombre de su selección. No necesita votar en cada carrera de elección.

Do not cross out, erase, or make extra marks anywhere on the ballot. This may invalidate your vote.
No raya sus selecciónes o no haga marcas extras; puede invalidar su voto.

Optional write-in
Voto opcional por escrito

or write-in
o por escrito:
Ann

To add a candidate, fill in the oval next to "write-in" and print the name clearly within the dashed-line box.
Si queire añadir un candidato, marque completamente el óvalo allado de "candidato por escrito" yescriba el nombre del candidato claramente adentro de la caja.

Optional: Confirmation
Opcional: confirmacione

N9
Confirmation Codes
Número de confirmaciónes
N9

175299

Favorite Tree *Árbol Favorito*	1st 1ra	2nd 2da	3rd 3ra	4th 4ta	5th 5ta
Cherry *el cerezo*					
Elm *el olmo*					
Maple *el arce*					
Oak *el roble*					
or write-in tree: *o por árbol escrito:*					

Favorite Forest Animal *Animal Arbolado Favorito*	1st 1ra	2nd 2da	3rd 3ra	4th 4ta
Owl *Búho*				
Rabbit *Conejo*				
Squirrel *Ardilla*				
or write-in animal: *o animal por escrito:*				

How many trees are on your property?
¿Cuanto árboles estan en su propiedad?

0
1-2
3-5
6-10
More than 10
Más de 10

Do you use less paper products than you did ten years ago?
¿Utiliza menos producto de papel que hace 10 años?

Yes *Sí*
No *No*

DO NOT detatch
NO separe

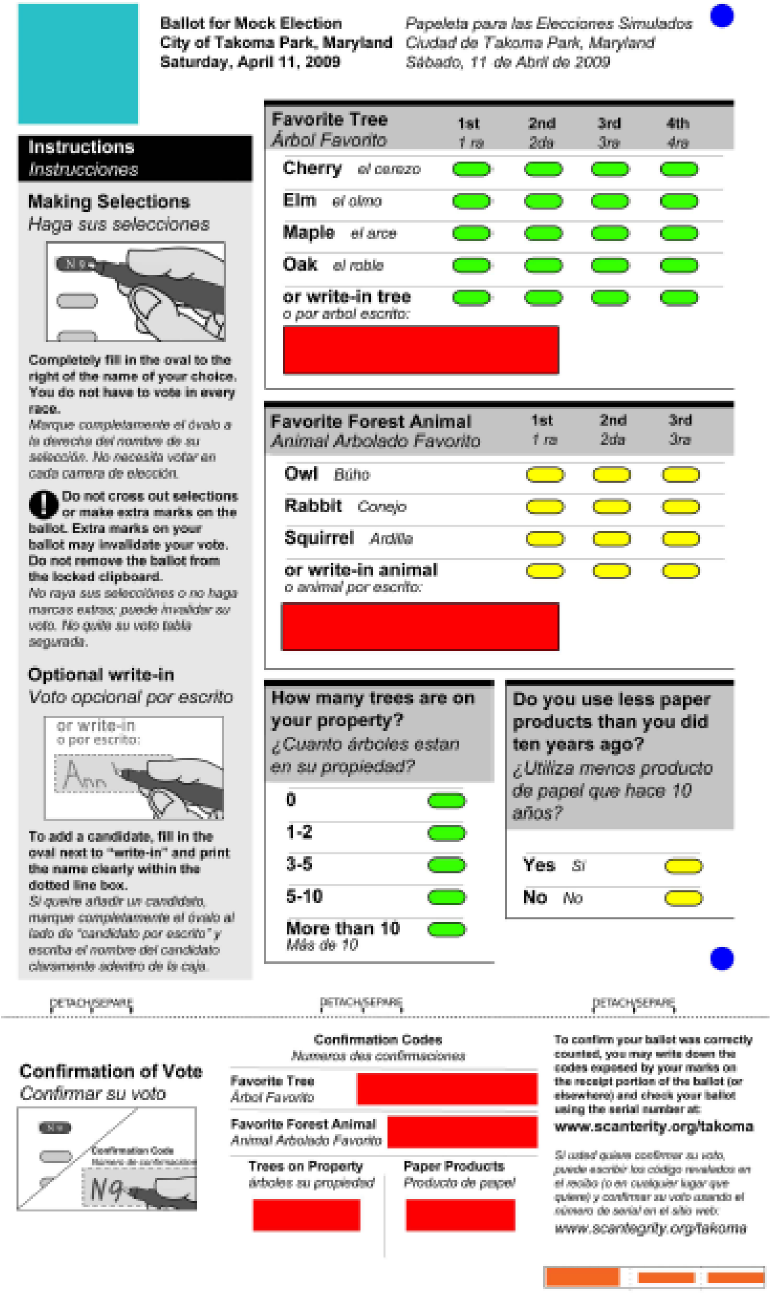
To confirm that your ballot was correctly counted, you may write down on this receipt (or elsewhere) the codes exposed by your marks. You'll be given this receipt after scanning. Then you enter the serial number (lower left) at the website: www.scantegrity.org/takoma
Si usted quiere confirmar su voto, puede escribir los código revalados en el recibo (o en cualquier lugar que quiere) y confirmar su voto usando el número de serial en el sitio web: www.scantegrity.org/takoma

Favorite Tree
Árbol Favorito
31 9C T4 CT 9C

Favorite Animal
Animal Favorito
HM TC YW Mk

Trees on Property *árboles su propiedad*	Paper Products *Producto de papel*
33	49

Figure 4.1: A picture of a ballot inside a locked clipboard used during the mock election. The locked clipboard was designed to defeat chain voting and ballot theft. Before giving the voter a ballot, the election judge locked the ballot in the clipboard. The scanning judge unlocked and scanned the ballot into the ballot box.



Ballot for Mock Election
City of Takoma Park, Maryland
Saturday, April 11, 2009

Papeleta para las Elecciones Simulados
Ciudad de Takoma Park, Maryland
Sábado, 11 de Abril de 2009

Instructions
Instrucciones

Making Selections
Haga sus selecciones

Completely fill in the oval to the right of the name of your choice. You do not have to vote in every race.
Marque completamente el óvalo a la derecha del nombre de su selección. No necesita votar en cada carrera de elección.

! Do not cross out selections or make extra marks on the ballot. Extra marks on your ballot may invalidate your vote. Do not remove the ballot from the locked clipboard.
No raya sus seleccíônes o no haga marcas extras; puede invalidar su voto. No quite su voto tabla segurada.

Optional write-in
Voto opcional por escrito

or write-in
o por escrito:

To add a candidate, fill in the oval next to "write-in" and print the name clearly within the dotted line box.
Si quiere añadir un candidato, marque completamente el óvalo al lado de "candidato por escrito" y escriba el nombre del candidato claramente adentro de la caja.

Favorite Tree *Árbol Favorito*	1st *1 ra*	2nd *2da*	3rd *3ra*	4th *4ra*
Cherry *el cerezo*	○	○	○	○
Elm *el olmo*	○	○	○	○
Maple *el arce*	○	○	○	○
Oak *el roble*	○	○	○	○
or write-in tree *o por arbol escrito:*	○	○	○	○

Favorite Forest Animal *Animal Arbolado Favorito*	1st *1 ra*	2nd *2da*	3rd *3ra*
Owl *Búho*	○	○	○
Rabbit *Conejo*	○	○	○
Squirrel *Ardilla*	○	○	○
or write-in animal *o animal por escrito:*	○	○	○

How many trees are on your property? *¿Cuanto árboles estan en su propiedad?*	
0	○
1-2	○
3-5	○
5-10	○
More than 10 *Más de 10*	○

Do you use less paper products than you did ten years ago? *¿Utiliza menos producto de papel que hace 10 años?*	
Yes *Sí*	○
No *No*	○

DETACH/SEPARE DETACH/SEPARE DETACH/SEPARE

Confirmation of Vote
Confirmar su voto

Confirmation Code
Numero de confirmacion

Confirmation Codes
Numeros des confirmaciones

Favorite Tree
Árbol Favorito

Favorite Forest Animal
Animal Arbolado Favorito

Trees on Property
árboles su propiedad

Paper Products
Producto de papel

To confirm your ballot was correctly counted, you may write down the codes exposed by your marks on the receipt portion of the ballot (or elsewhere) and check your ballot using the serial number at:
www.scanterity.org/takoma

Si usted quiere confirmar su voto, puede escribir los código revelados en el recibo (o en cualquier lugar que quiere) y confirmar su voto usando el número de serial en el sitio web:
www.scantegrity.org/takoma

Figure 4.2: The ballot used during the mock election. There are two contests and two questions on this ballot. The contests each use an IRV tally, and the questions use a plurality tally. The detachable receipt is on the bottom portion of the ballot. Note that the ballot serial, left chit serial, and right chit serial numbers are next to each other on this ballot design.

such ballots. We used a similar procedure if the voter unintentionally spoiled a ballot (e.g., by marking the wrong choice).

After marking the ballot, the voter proceeded to the scanning table. A poll worker unlocked the ballot from the locked clipboard and scanned the ballot. Looking at a touch-screen display connected to the scanner, the voter confirmed that the ballot was scanned. Without showing the voter's ballot choices, the touch-screen display warned the voter if the scanner detected any over- or undervoted question. At this point, the voter could either return to the voting area with the ballot or cast the ballot by pressing the cast button on the display. The poll worker then tore off the chit and gave it to the voter, and dropped the ballot into the ballot box. Throughout the scanning process, a privacy sleeve hid the ballot choices.

The chit provided instructions on how the voter could optionally verify her vote online after polls closed.

4.1.2 Research Protocols

Any consenting adult who showed up was permitted to vote. Also at the request of Takoma Park—to encourage children to become involved in voting and new voting technology—assenting children from 12 to 17 years old were also permitted to vote with parental consent. We advertised the event through e-mail, webpages, local TV, and the Takoma Park Newsletter. Despite the rain, 105 people signed consent forms.

Sitting in the polling room in the place reserved for official observers, two unobtrusive observers watched as many voters as possible and filled out voter observation sheets. Each observer recorded the time an observed voter spent from receiving a ballot to casting it. Each observer also noted how many times the voter spoiled a ballot, requested or received assistance from a poll worker, or appeared confused.

As each voter left the polling room, a researcher asked the voter if she would be willing to fill out a questionnaire. If the answer was yes, the researcher handed the voter a conventional clipboard with two two-sided questionnaires: a voter field test questionnaire and a demographics questionnaire. Form numbers linked the field test and demographics questionnaires filled out by each voter.

As the voter returned the clipboard and exited the polling area, a researcher asked the voter if she would be willing to return at 3pm that day for a one-hour focus group. For each willing voter, the researcher wrote down a telephone number and the demographics form number. The plan was to call eight of the willing voters, reflecting a diverse sample of voters as determined solely from the demographics form. However, given that only 12 of the 80 voters filling out questionnaires agreed to participate in a focus group, we invited all twelve willing voters, of whom four participated.

We also conducted a separate one-hour focus group for all six poll workers as soon as possible after polls closed. Each poll worker also filled out a poll worker field test questionnaire and demographics form.

Voters could visit the verification website after polls closed. After providing consent and verifying their votes online, they were invited to fill out an online verification questionnaire and a short demographics form.

Aside from the consent form and list of telephone numbers on the focus group sign-up sheet, we did not collect any personal identifying information.

Originally, we had planned to link each voter's demographics questionnaire to her observation sheet and ballot (and thereby to her verification questionnaire). Ultimately, we decided not to do so to avoid interfering with the election process and avoid creating the appearance of violating ballot privacy. Instead, we added a second short demographics questionnaire to the online verification experience.

To help election judges learn how to operate the system, Takoma Park judges and Scantegrity team members worked side-by-side during the mock election. By contrast, in the binding election in November, poll workers operated the system entirely by themselves.

4.1.3 Research Instruments

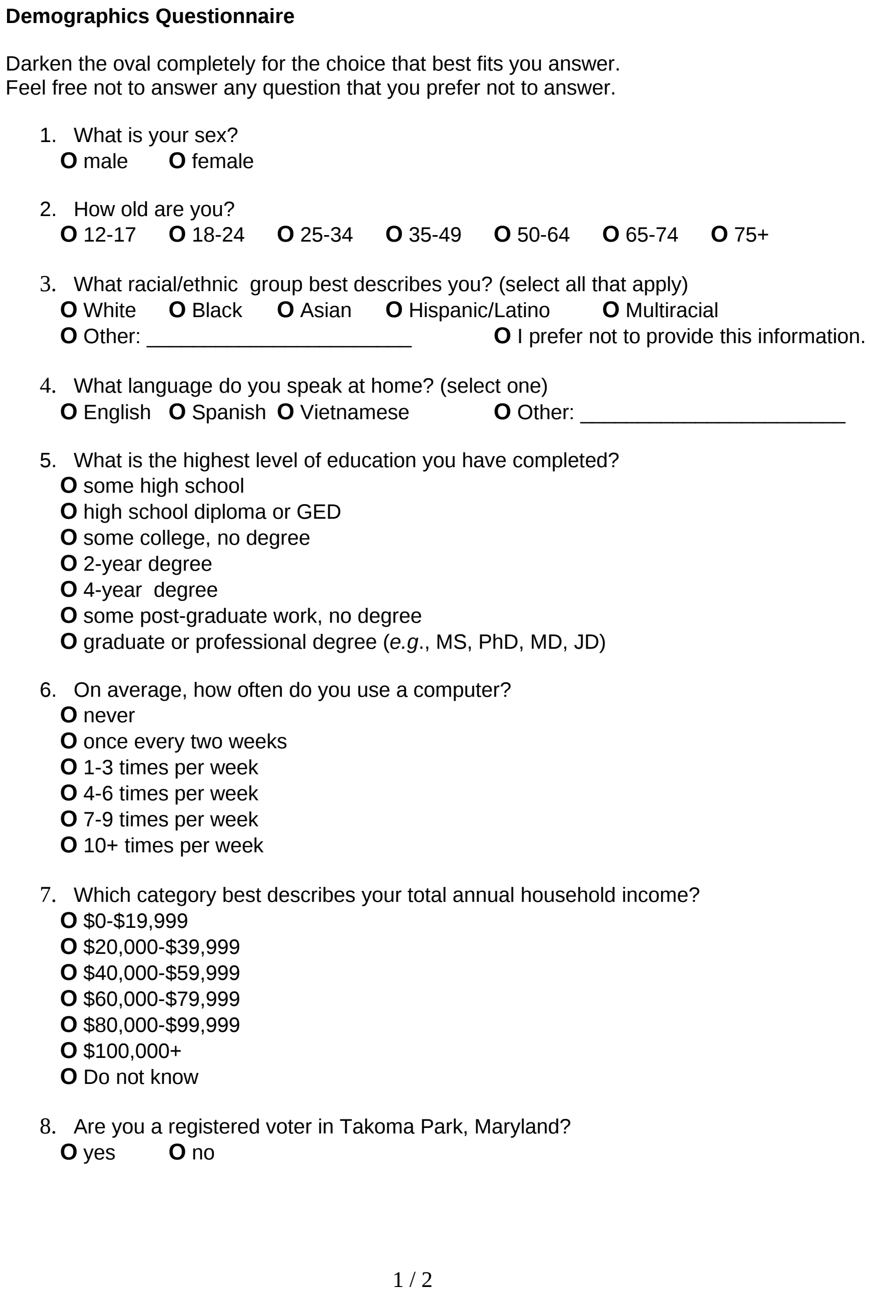
We gave three forms to participants: a demographics form, a voter field study questionnaire, and an election judge field study questionnaire.

Demographics Form

We gave voters and election officials the same demographics form, consisting of 15 questions on standard demographic information and information about prior experience with different types of voting systems. The questions asked respondents about their sex, age, race, languages spoken, education level, computer usage, participation in previous elections, whether any mistakes had been made while voting in previous elections, previous voting systems used, physical challenges, annual household income, and political opinions. See Figures 4.3 and 4.4 for the demographics form.

Voter Field Study Questionnaire

The voter questionnaire comprised 27 questions. The first 19 used a 7-point Likert (strongly disagree to strongly agree) scale, and included a "not applicable" option. The remaining questions asked respondents how many times they made mistakes, whether they attempted to audit ballot printing, whether they asked for assistance, whether they had any difficulties voting, and whether they had comments about the process, and ended with four 7-point Likert scale questions comparing Scantegrity to traditional optical scan systems. Figures 4.5 and 4.6 show the voter questionnaires used during the mock election.



Demographics Questionnaire

Darken the oval completely for the choice that best fits you answer.
Feel free not to answer any question that you prefer not to answer.

1. What is your sex?
   O male **O** female

2. How old are you?
   O 12-17 **O** 18-24 **O** 25-34 **O** 35-49 **O** 50-64 **O** 65-74 **O** 75+

3. What racial/ethnic group best describes you? (select all that apply)
   O White **O** Black **O** Asian **O** Hispanic/Latino **O** Multiracial
   O Other: _______________________ **O** I prefer not to provide this information.

4. What language do you speak at home? (select one)
   O English **O** Spanish **O** Vietnamese **O** Other: _______________________

5. What is the highest level of education you have completed?
   O some high school
   O high school diploma or GED
   O some college, no degree
   O 2-year degree
   O 4-year degree
   O some post-graduate work, no degree
   O graduate or professional degree (*e.g.*, MS, PhD, MD, JD)

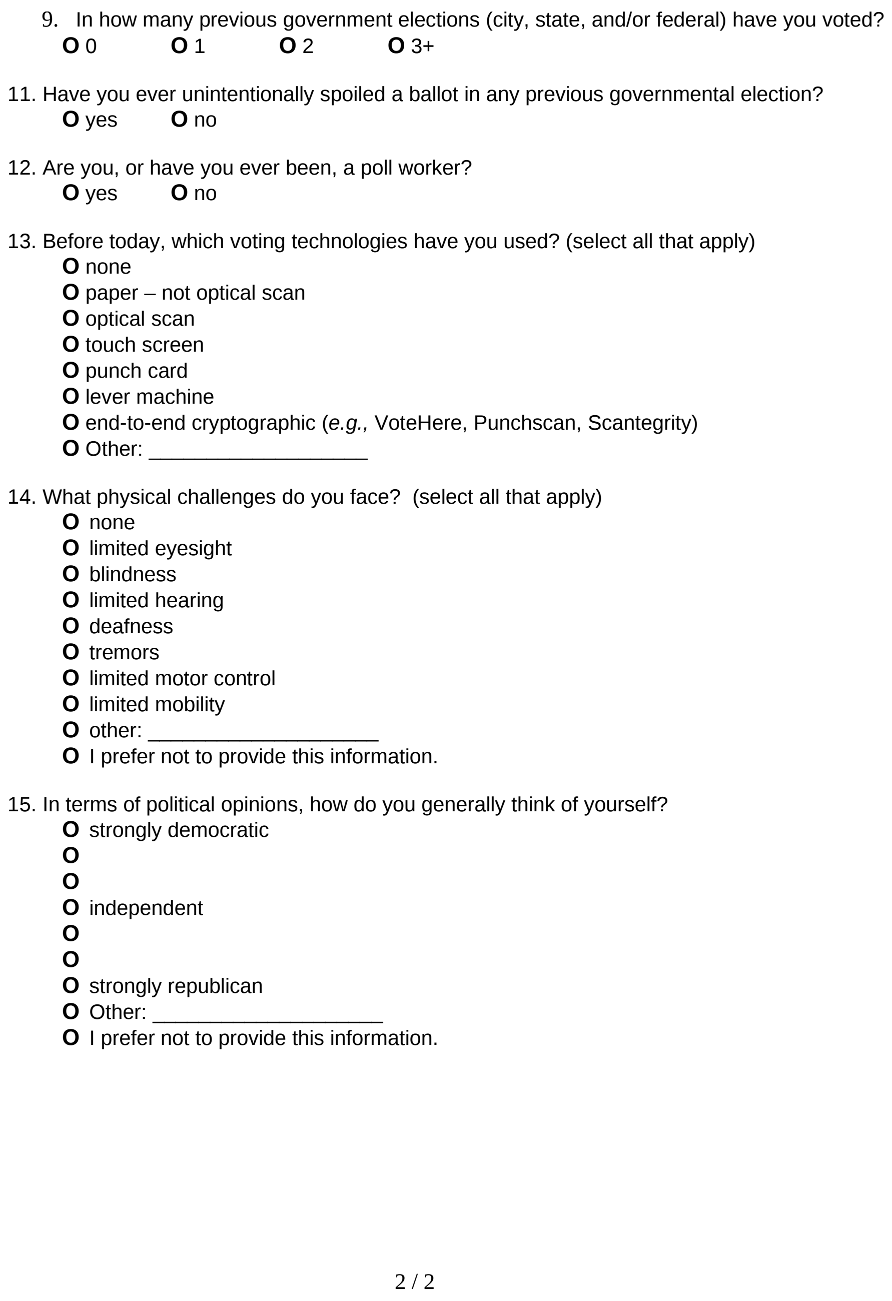
6. On average, how often do you use a computer?
   O never
   O once every two weeks
   O 1-3 times per week
   O 4-6 times per week
   O 7-9 times per week
   O 10+ times per week

7. Which category best describes your total annual household income?
   O $0-$19,999
   O $20,000-$39,999
   O $40,000-$59,999
   O $60,000-$79,999
   O $80,000-$99,999
   O $100,000+
   O Do not know

8. Are you a registered voter in Takoma Park, Maryland?
   O yes **O** no

1 / 2

Figure 4.3: Page 1 of the demographics form used during the mock election.



9. In how many previous government elections (city, state, and/or federal) have you voted?
○ 0 ○ 1 ○ 2 ○ 3+

11. Have you ever unintentionally spoiled a ballot in any previous governmental election?
○ yes ○ no

12. Are you, or have you ever been, a poll worker?
○ yes ○ no

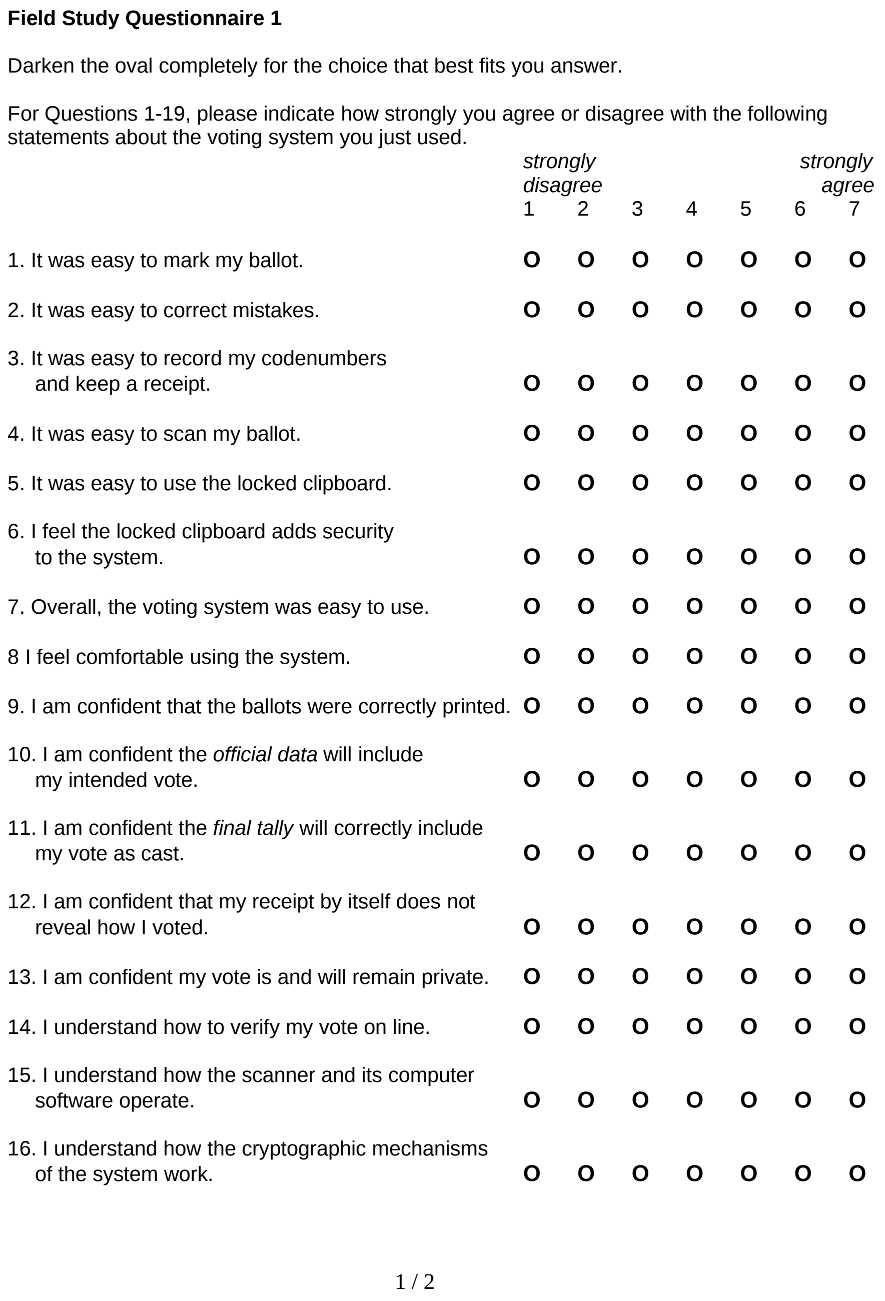
13. Before today, which voting technologies have you used? (select all that apply)
○ none
○ paper – not optical scan
○ optical scan
○ touch screen
○ punch card
○ lever machine
○ end-to-end cryptographic (*e.g.,* VoteHere, Punchscan, Scantegrity)
○ Other: ___________________

14. What physical challenges do you face? (select all that apply)
○ none
○ limited eyesight
○ blindness
○ limited hearing
○ deafness
○ tremors
○ limited motor control
○ limited mobility
○ other: ____________________
○ I prefer not to provide this information.

15. In terms of political opinions, how do you generally think of yourself?
○ strongly democratic
○
○
○ independent
○
○
○ strongly republican
○ Other: ____________________
○ I prefer not to provide this information.

2 / 2

Figure 4.4: Page 2 of the demographics form used during the mock election.



Field Study Questionnaire 1

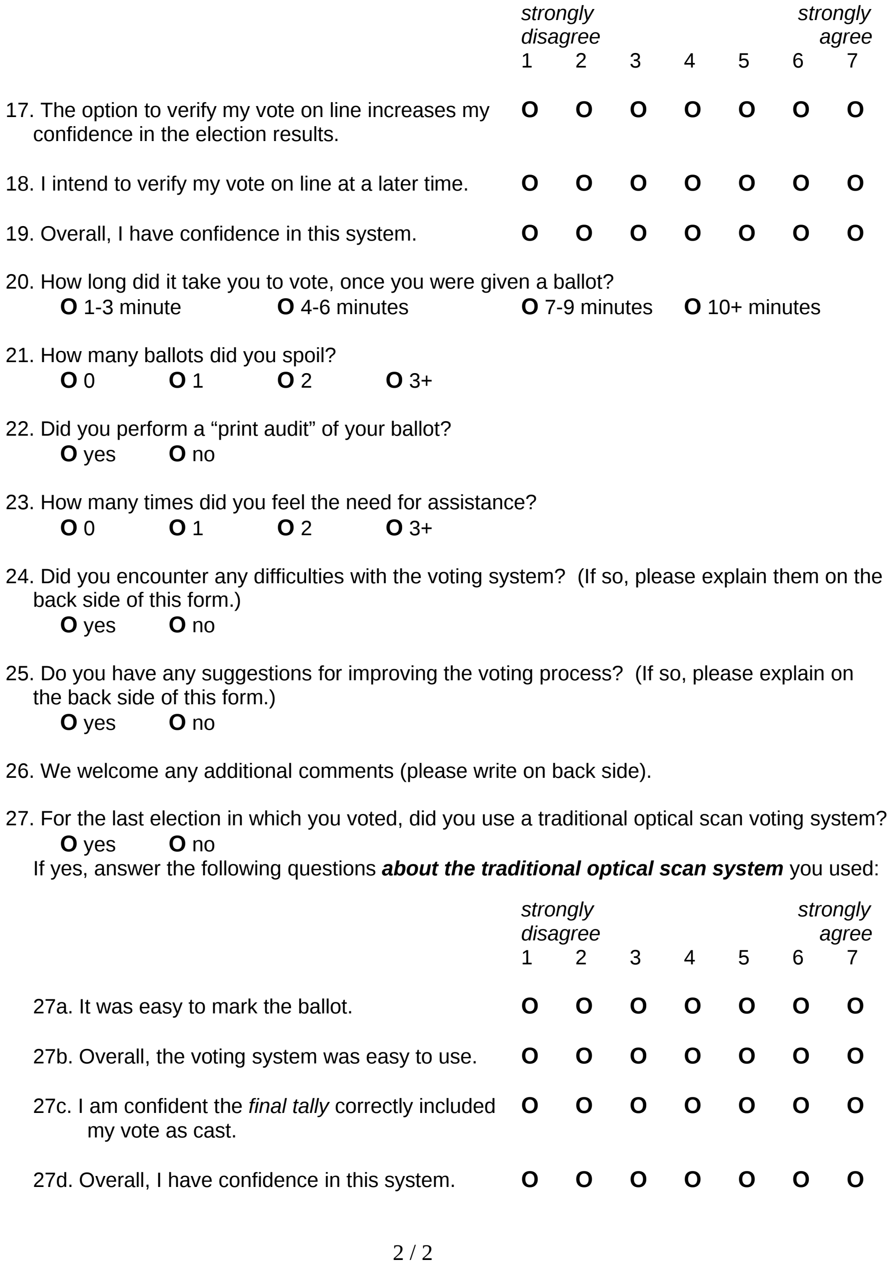
Darken the oval completely for the choice that best fits you answer.

For Questions 1-19, please indicate how strongly you agree or disagree with the following statements about the voting system you just used.

	strongly disagree 1	2	3	4	5	6	*strongly agree* 7
1. It was easy to mark my ballot.	○	○	○	○	○	○	○
2. It was easy to correct mistakes.	○	○	○	○	○	○	○
3. It was easy to record my codenumbers and keep a receipt.	○	○	○	○	○	○	○
4. It was easy to scan my ballot.	○	○	○	○	○	○	○
5. It was easy to use the locked clipboard.	○	○	○	○	○	○	○
6. I feel the locked clipboard adds security to the system.	○	○	○	○	○	○	○
7. Overall, the voting system was easy to use.	○	○	○	○	○	○	○
8 I feel comfortable using the system.	○	○	○	○	○	○	○
9. I am confident that the ballots were correctly printed.	○	○	○	○	○	○	○
10. I am confident the *official data* will include my intended vote.	○	○	○	○	○	○	○
11. I am confident the *final tally* will correctly include my vote as cast.	○	○	○	○	○	○	○
12. I am confident that my receipt by itself does not reveal how I voted.	○	○	○	○	○	○	○
13. I am confident my vote is and will remain private.	○	○	○	○	○	○	○
14. I understand how to verify my vote on line.	○	○	○	○	○	○	○
15. I understand how the scanner and its computer software operate.	○	○	○	○	○	○	○
16. I understand how the cryptographic mechanisms of the system work.	○	○	○	○	○	○	○

1 / 2

Figure 4.5: Page 1 of the voter questionnaire form used during the mock election.



	strongly disagree						*strongly agree*
	1	2	3	4	5	6	7
17. The option to verify my vote on line increases my confidence in the election results.	○	○	○	○	○	○	○
18. I intend to verify my vote on line at a later time.	○	○	○	○	○	○	○
19. Overall, I have confidence in this system.	○	○	○	○	○	○	○

20. How long did it take you to vote, once you were given a ballot?
○ 1-3 minute ○ 4-6 minutes ○ 7-9 minutes ○ 10+ minutes

21. How many ballots did you spoil?
○ 0 ○ 1 ○ 2 ○ 3+

22. Did you perform a "print audit" of your ballot?
○ yes ○ no

23. How many times did you feel the need for assistance?
○ 0 ○ 1 ○ 2 ○ 3+

24. Did you encounter any difficulties with the voting system? (If so, please explain them on the back side of this form.)
○ yes ○ no

25. Do you have any suggestions for improving the voting process? (If so, please explain on the back side of this form.)
○ yes ○ no

26. We welcome any additional comments (please write on back side).

27. For the last election in which you voted, did you use a traditional optical scan voting system?
○ yes ○ no
If yes, answer the following questions ***about the traditional optical scan system*** you used:

	strongly disagree						*strongly agree*
	1	2	3	4	5	6	7
27a. It was easy to mark the ballot.	○	○	○	○	○	○	○
27b. Overall, the voting system was easy to use.	○	○	○	○	○	○	○
27c. I am confident the *final tally* correctly included my vote as cast.	○	○	○	○	○	○	○
27d. Overall, I have confidence in this system.	○	○	○	○	○	○	○

2 / 2

Figure 4.6: Page 2 of the voter questionnaire form used during the mock election.

Election Judge Field Study Questionnaire

The election judge questionnaire featured twenty 7-point Likert scale questions, which asked whether the system was easy to administer, whether it was easy for voters to mark ballots, and other related statements. Like the voter questionnaire, the rest of the questionnaire asked for comments and for the judges to rate Scantegrity against experiences with traditional optical scan systems. Figures 4.7 and 4.8 show the election judge questionnaire.

4.2 Results

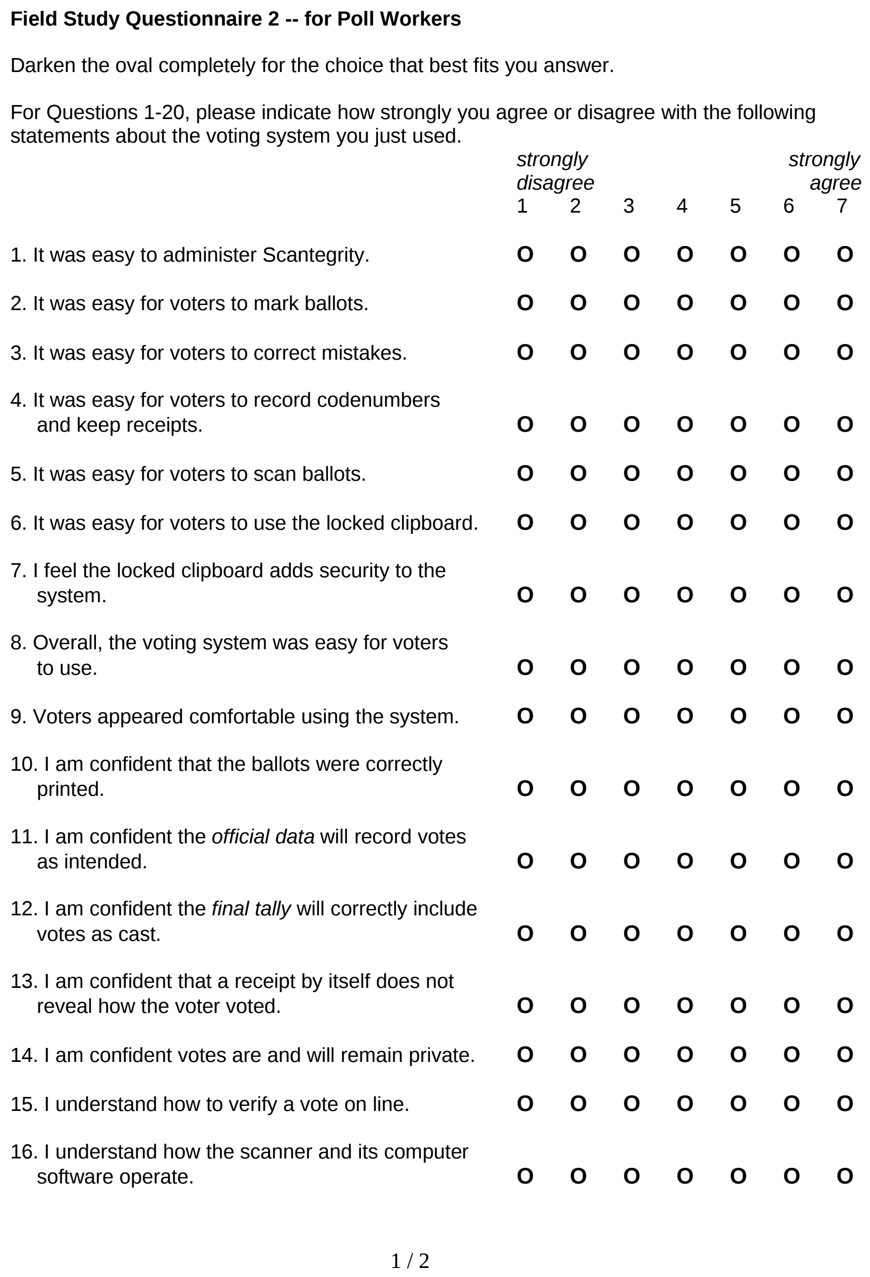
This section summarizes data collected from our research instruments, including the voter demographics questionnaire, observation sheets, voter field test questionnaire, online voter demographics and verification questionnaires, and voter and poll worker focus groups.

4.2.1 Voter Demographics

Figures 4.9 and 4.10 summarize voter characteristics of the 80 voters who filled out paper demographics questionnaires. These voters were not representative of the Takoma Park voting population. They had high family incomes and were highly educated, frequent computer users, mostly 50 to 64 years old, motivated, and able to get to the mock election on their own.

4.2.2 Unobtrusive Observations

Figure 4.11 summarizes observations made by two unobtrusive observers watching 53 of the voters. The main difficulty was the length of time it took to vote, averaging about 8 minutes from the time a voter received a ballot to the time the voter cast the ballot (not including



Field Study Questionnaire 2 -- for Poll Workers

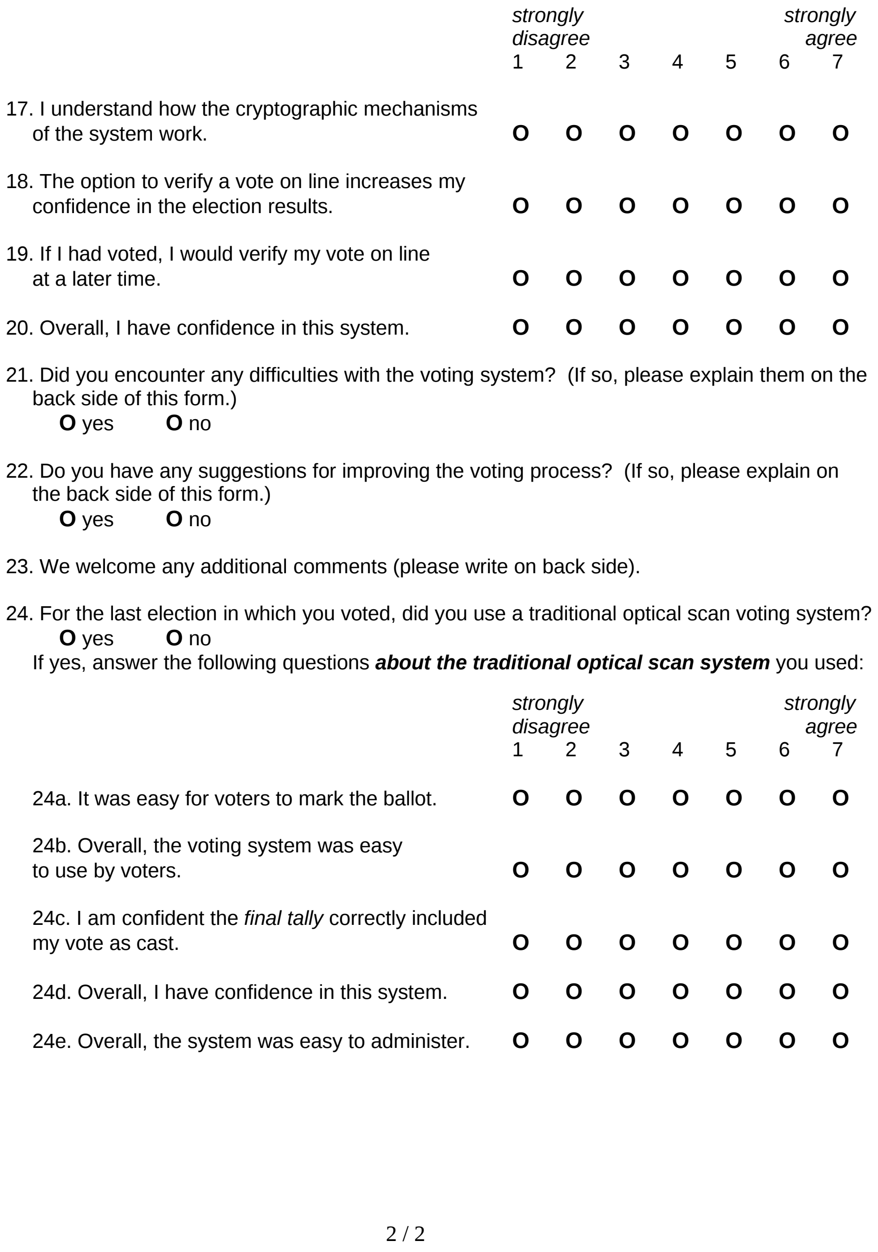
Darken the oval completely for the choice that best fits you answer.

For Questions 1-20, please indicate how strongly you agree or disagree with the following statements about the voting system you just used.

	strongly disagree 1	2	3	4	5	6	*strongly agree* 7
1. It was easy to administer Scantegrity.	○	○	○	○	○	○	○
2. It was easy for voters to mark ballots.	○	○	○	○	○	○	○
3. It was easy for voters to correct mistakes.	○	○	○	○	○	○	○
4. It was easy for voters to record codenumbers and keep receipts.	○	○	○	○	○	○	○
5. It was easy for voters to scan ballots.	○	○	○	○	○	○	○
6. It was easy for voters to use the locked clipboard.	○	○	○	○	○	○	○
7. I feel the locked clipboard adds security to the system.	○	○	○	○	○	○	○
8. Overall, the voting system was easy for voters to use.	○	○	○	○	○	○	○
9. Voters appeared comfortable using the system.	○	○	○	○	○	○	○
10. I am confident that the ballots were correctly printed.	○	○	○	○	○	○	○
11. I am confident the *official data* will record votes as intended.	○	○	○	○	○	○	○
12. I am confident the *final tally* will correctly include votes as cast.	○	○	○	○	○	○	○
13. I am confident that a receipt by itself does not reveal how the voter voted.	○	○	○	○	○	○	○
14. I am confident votes are and will remain private.	○	○	○	○	○	○	○
15. I understand how to verify a vote on line.	○	○	○	○	○	○	○
16. I understand how the scanner and its computer software operate.	○	○	○	○	○	○	○

1 / 2

Figure 4.7: Page 1 of the election judge questionnaire form used during the mock election.



	strongly disagree						*strongly agree*
	1	2	3	4	5	6	7
17. I understand how the cryptographic mechanisms of the system work.	○	○	○	○	○	○	○
18. The option to verify a vote on line increases my confidence in the election results.	○	○	○	○	○	○	○
19. If I had voted, I would verify my vote on line at a later time.	○	○	○	○	○	○	○
20. Overall, I have confidence in this system.	○	○	○	○	○	○	○

21. Did you encounter any difficulties with the voting system? (If so, please explain them on the back side of this form.)
○ yes ○ no

22. Do you have any suggestions for improving the voting process? (If so, please explain on the back side of this form.)
○ yes ○ no

23. We welcome any additional comments (please write on back side).

24. For the last election in which you voted, did you use a traditional optical scan voting system?
○ yes ○ no
If yes, answer the following questions ***about the traditional optical scan system*** you used:

	strongly disagree						*strongly agree*
	1	2	3	4	5	6	7
24a. It was easy for voters to mark the ballot.	○	○	○	○	○	○	○
24b. Overall, the voting system was easy to use by voters.	○	○	○	○	○	○	○
24c. I am confident the *final tally* correctly included my vote as cast.	○	○	○	○	○	○	○
24d. Overall, I have confidence in this system.	○	○	○	○	○	○	○
24e. Overall, the system was easy to administer.	○	○	○	○	○	○	○

2 / 2

Figure 4.8: Page 2 of the election judge questionnaire form used during the mock election.

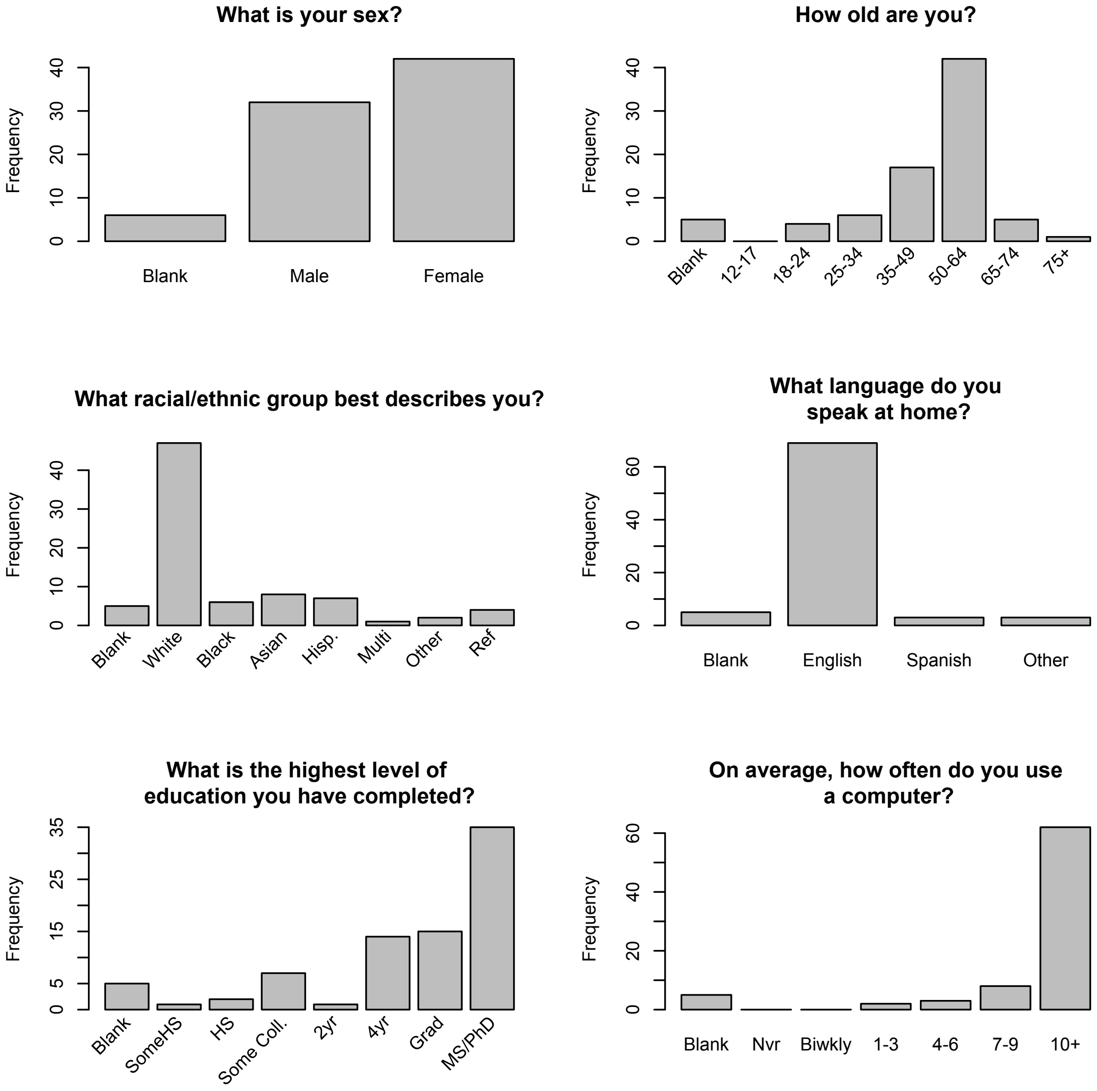
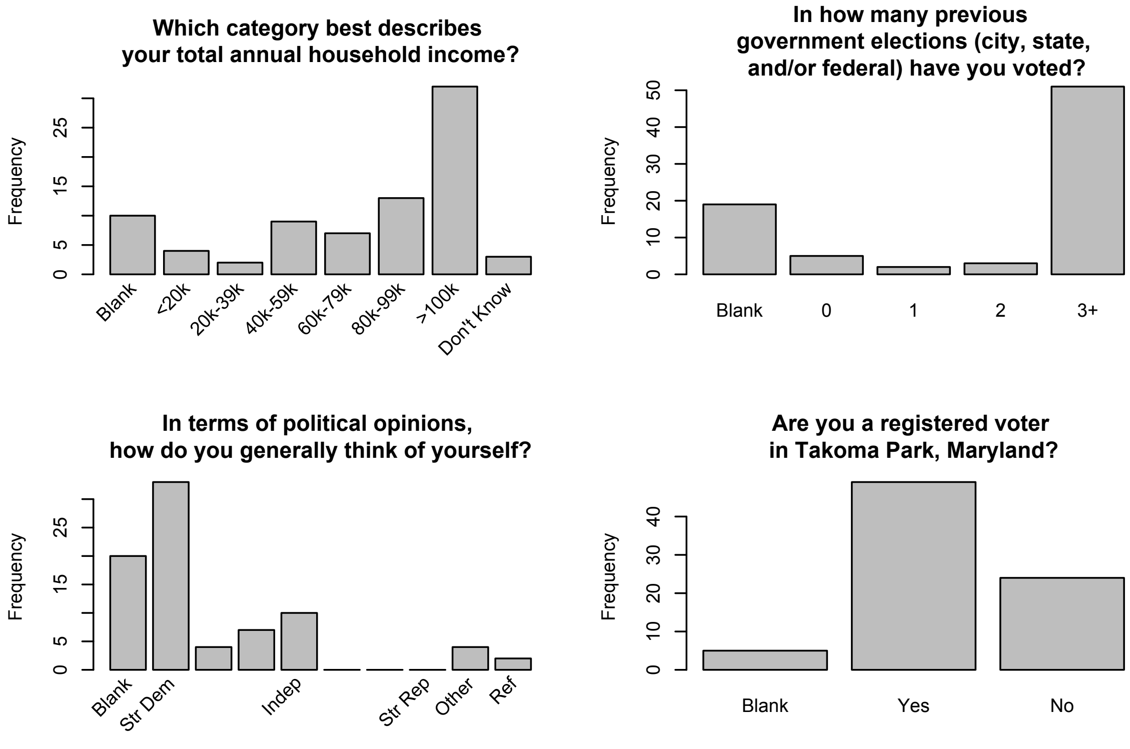


Figure 4.9: Summary and comparison of voter demographics from 80 responses to a paper questionnaire filled out by voters immediately after voting.


Which category best describes your total annual household income?

In how many previous government elections (city, state, and/or federal) have you voted?

In terms of political opinions, how do you generally think of yourself?

Are you a registered voter in Takoma Park, Maryland?

Figure 4.10: Continuation of figure 4.9. Summary and comparison of voter demographics from 80 responses to a paper questionnaire filled out by voters immediately after voting.

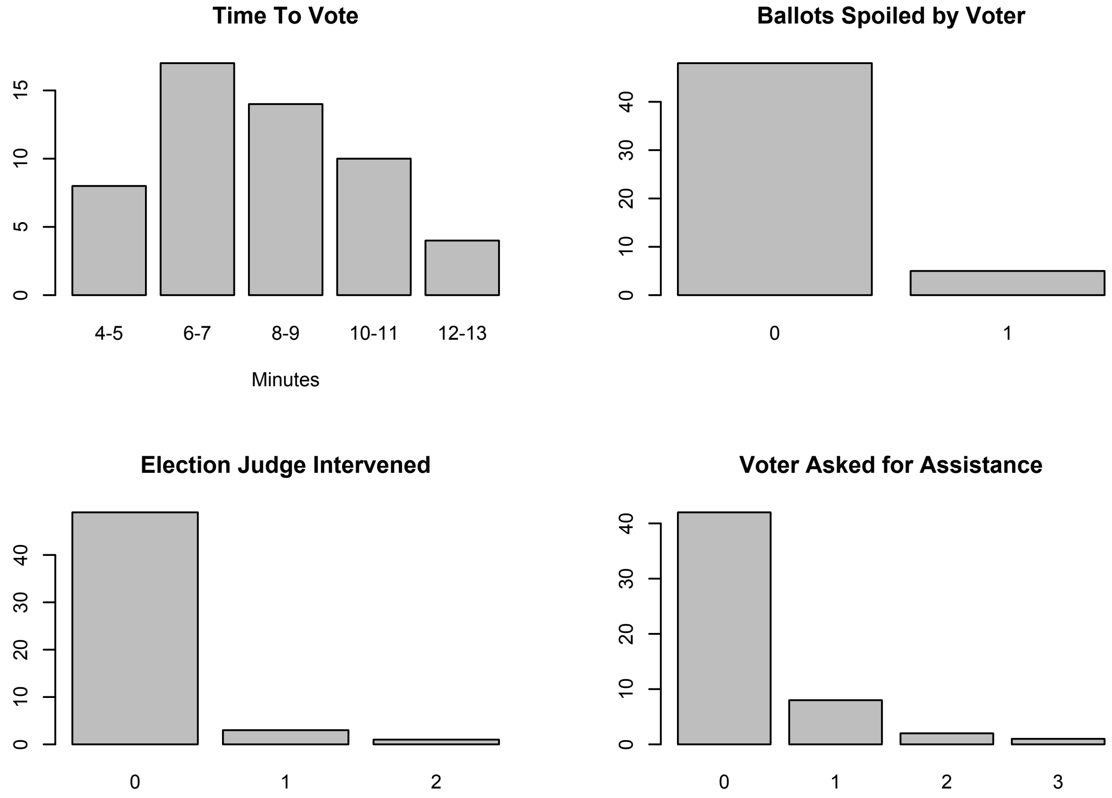


Figure 4.11: Summary of data from unobtrusive observations of 53 voters.

time for check-in or instructions given before voter received a ballot). Much of the time was observed to be at the scanner table.

When voters asked for assistance and/or poll workers intervened, it was typically because the voter did not know what to do after marking the ballot or what to do upon spoiling a ballot.

4.2.3 Voter Field Test Survey

Figures 4.12, 4.13, 4.14, and 4.15 summarize data collected from 80 field test questionnaires filled out by voters immediately after casting their ballots. We include all responses, even though it was apparent (from implausible answers to questions about ease of correcting

#	Question	1	2	3	4	5	6	7
1	I was able to complete the verification process.	0	1	0	0	0	2	4
2	I verified that my intended votes were correctly recorded as cast.	1	1	0	0	0	2	3
3	The verification system was easy to use.	1	0	0	0	0	1	5
4	I feel comfortable using the verification system.	1	0	0	0	0	3	3
5	I am confident the official data includes my intended vote.	0	1	0	2	1	1	2
6	I am confident the final tally includes my intended vote.	0	1	0	3	0	1	2
7	I am confident my vote is and will remain private.	0	1	0	1	2	2	1
8	The online verification increased my confidence in the election results.	1	2	1	1	1	0	1
9	I understand how the online verification system works.	0	3	1	0	0	1	2
10	I have confidence in the online verification system.	0	1	0	4	1	0	1
11	Overall, I have confidence in Scantegrity.	0	0	1	3	2	0	1

Table 4.1: Summary of all seven responses from the online verification questionnaire (1 = strongly disagree, 7 = strongly agree). For example, four respondents strongly agreed with the first statement.

errors and understanding of cryptographic details) that three respondents had likely reversed the seven-point Likert scale.

4.2.4 Online Voter Verification Survey

As of April 15, thirty-one voters verified their votes online. Seven of these voters completed the associated online questionnaire. Table 4.1 summarizes the responses from these seven voters.

4.2.5 Voter and Poll Worker Focus Groups

Four voters participated in the voter focus group. These came from the 12 voters who stated they might be available to participate, all of whom were invited. These four voters were not

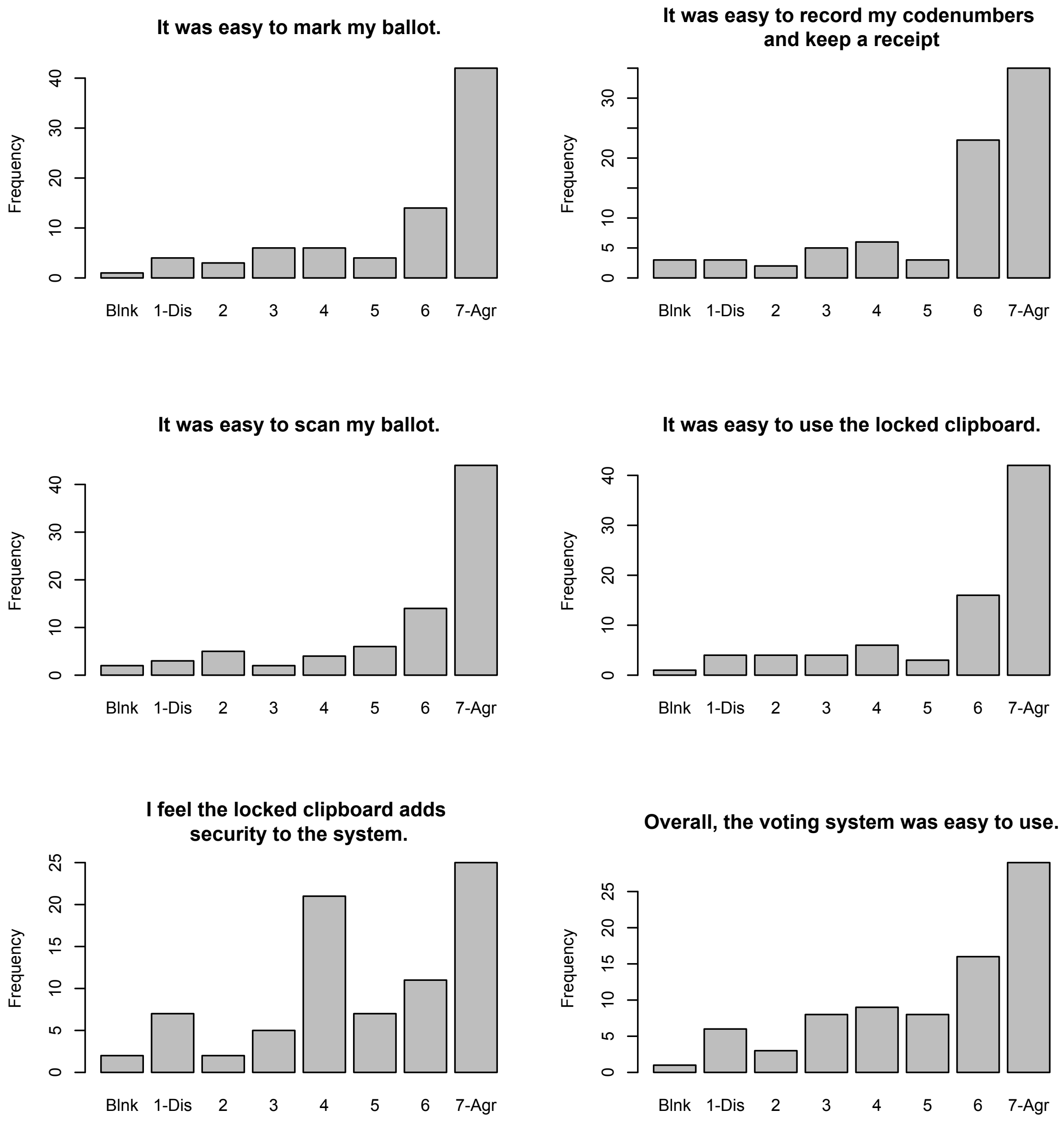
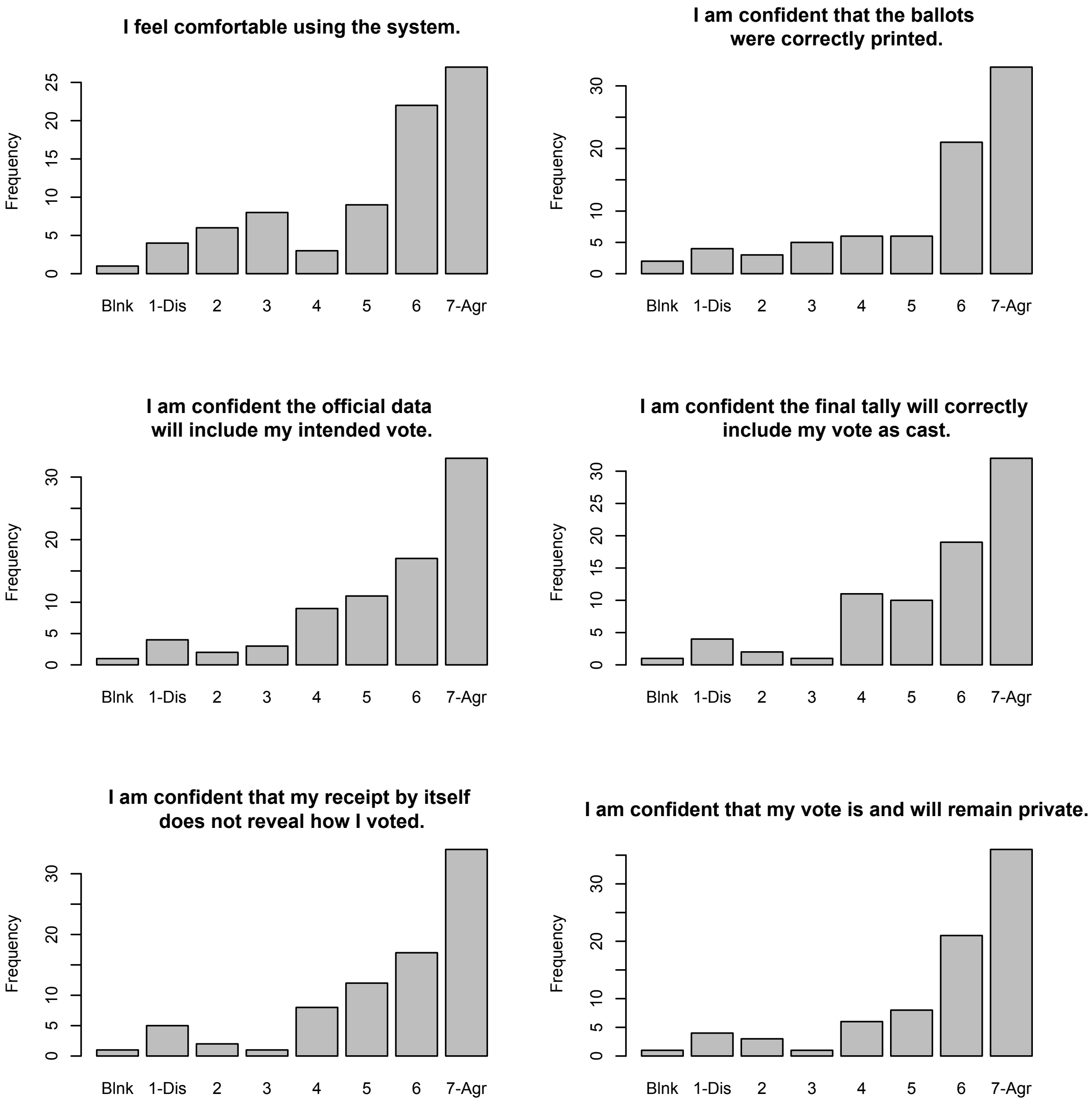


Figure 4.12: Summary of responses to questions 1, 3, 4, 5, 6, and 7 from a paper questionnaire about Scantegrity filled out by 80 voters immediately after voting.



I feel comfortable using the system.

I am confident that the ballots were correctly printed.

I am confident the official data will include my intended vote.

I am confident the final tally will correctly include my vote as cast.

I am confident that my receipt by itself does not reveal how I voted.

I am confident that my vote is and will remain private.

Figure 4.13: Summary of responses to questions 8 through 13 from a paper questionnaire about Scantegrity filled out by 80 voters immediately after voting.

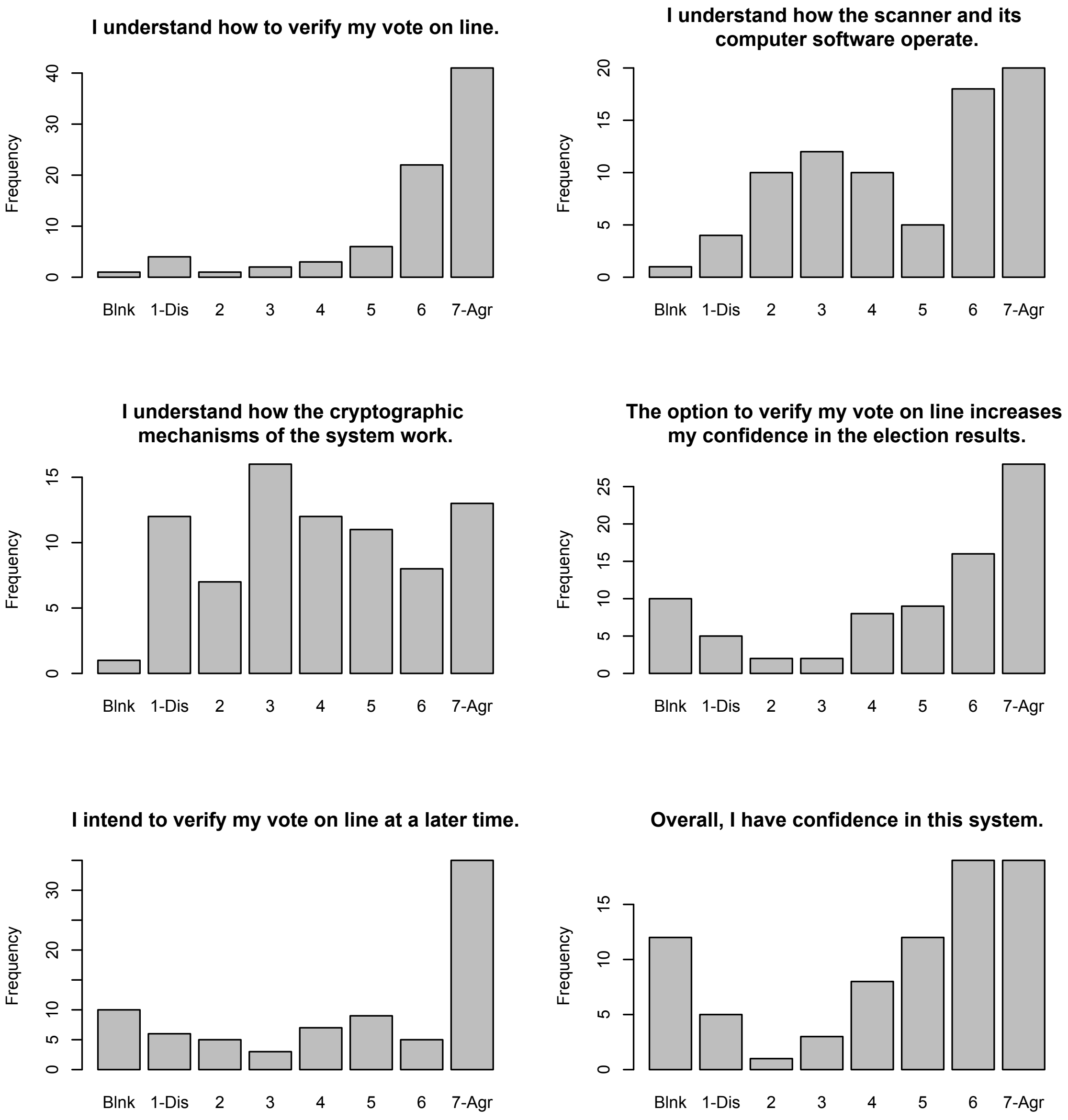


Figure 4.14: Summary of responses to questions 14 through 19 from a paper questionnaire about Scantegrity filled out by 80 voters immediately after voting.

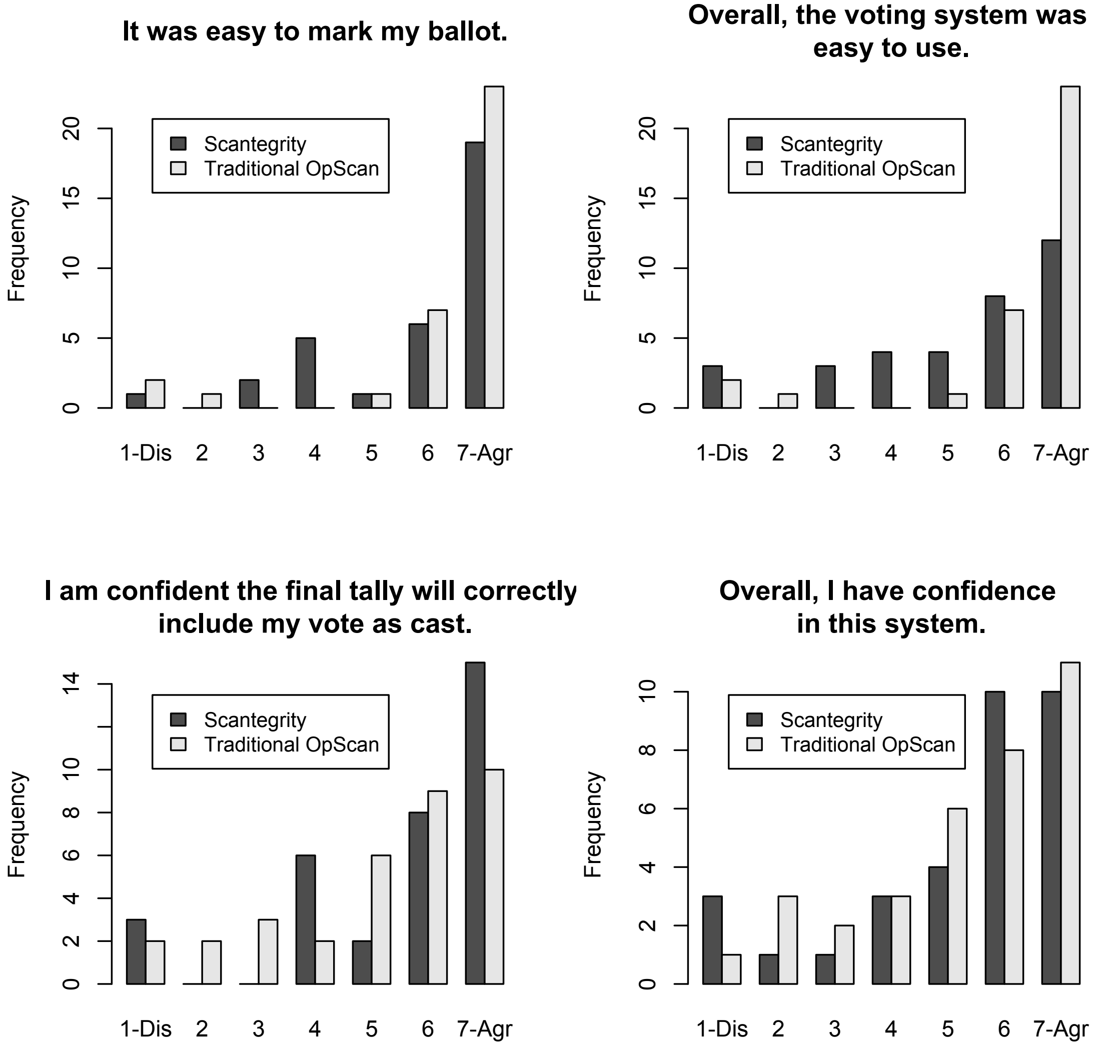


Figure 4.15: Summary of 31 responses to questions about Scantegrity, and a comparison to answers from those same responders about traditional optical scan systems based on their recollection of their last experience with an optical scan system.

representative of the Takoma Park voting population: they were involved with municipal functions and some had helped bring voters to previous elections.

All six Takoma Park poll workers participated in the poll worker focus group. Each was experienced and had worked in previous elections in Takoma Park. None are part of the Scantegrity team.

Because both groups expressed similar thoughts, we now summarize the main comments from both groups together, as reported by the moderator:

1. The process took too much time.
2. Providing instructions in one chunk at beginning was overwhelming.
3. The instructions were too complex, and there was too much explaining.
4. Although the voters in the focus group did not experience difficulties voting, some wondered if other voters in Takoma Park might experience difficulties writing down confirmation numbers and verifying their votes online.
5. Vote casting at the scanning table took too much time.
6. Some poll workers disliked that a poll worker handled the ballot during scanning.
7. The scanner was finicky.
8. During scanning, the poll workers liked the feedback of seeing light on a flash drive blink, suggesting that the ballot was read.
9. The locked clipboard added time and complexity but did not significantly increase security.
10. Make the special pens available only in the voting area.

11. Poll workers felt that they should have been more in charge, especially of the flow of voters around the room.

12. Poll workers felt that the process could be sped up to make it viable for the binding election.

Finally, the moderator emphasized, “It is critical that all instructions are tested ahead of time on a range of people representative of the wider Takoma Park population to ensure they are clear and understandable,” and “Translations into other languages must also be tested.”

4.3 Discussion

The two main issues were that the process was too slow (taking about 8 minutes to vote on average) and that many voters found the instructions somewhat complicated. Much of the delay was caused by the scanning process and lengthy instructions given to voters. Fortunately, these problems are solvable through process simplification and improvement, better scanners, and careful human-factors testing.

The Scantegrity team is comprised of experts in cryptography and computer security who have focused heavily on technical innovations and engineering solutions to make high-integrity elections possible. Although there has been tremendous simplification of Chaum’s ideas from SureVote (based on visual cryptography) through Punchscan (with two-sheet ballots) to Scantegrity (an add-on to optical scan voting), the team had spent relatively little effort on testing and perfecting the human-factors details of the voting process, especially when carried out by typical voters. Some Mock election voters were enthusiastic about the security features of Scantegrity, but most seemed not to care much about security, focusing primarily on the physical process of receiving a ballot, marking the ballot, and scanning the ballot. While such voter reactions are well known from the social science literature, it was nevertheless a dramatic learning experience to witness these reactions firsthand.

Although the Mock election voters and participants in the voter survey group were not typical Takoma Park voters (many were self-selected as having an interest in the voting system to be used by the city, and some were just there to participate in Arbor Day), they provided useful feedback and expressed awareness of potential issues that might affect other voters.

Factors affecting the slow voting process included lengthy instructions, redundant instructions, instructions for optional steps, use of the locked clipboard, writing down confirmation numbers, tearing off the ballot chit, difficulty of correcting mistakes (for the few who unintentionally spoiled ballots), checking for over- and under-votes at the scanner touch screen, and a slow, finicky scanner.

Our single inexpensive scanner caused significant problems. Ballots had to be inserted in a particular orientation. If they went in at too much of an angle, a corner could be unread. The scanning process created delays. Some voters seemed confused that the touch screen did not show how they voted, but only for each race whether the race was over- or under-voted. After the voter pressed "cast," the process of feeding the scanned ballot into a privacy sleeve and dropping the ballot into a large ballot box was clumsy. These equipment, implementation, and process problems would create severe difficulties in an election with over 2,000 voters, but they can be fixed.

The locked clipboard worked poorly. It complicated and slowed down the process, made it difficult to drop ballots into the scanner, and added weight. Most voters felt it did not enhance security, despite its purpose of making it difficult to steal or swap ballots. At the scanning table, several voters mistakenly ripped their ballots off the locked clipboard.

Technically, any ballot with torn locking hole was supposed to be invalid, but for simplicity this rule was not enforced.

Some elderly voters commented that they had difficulty reading the confirmation codes. Three voters reported that some confirmation numbers blurred, especially if rubbed heavily, and in one case the ballot paper deteriorated.

Although we had intended to use a reduced character set for confirmation numbers to avoid easily confused letters (e.g., “i” and “j”), this intention was not implemented, causing confusion for a few voters.

On a positive note, marking the ballot with revealing ink produced perfectly darkened ovals: because there was no reactive ink outside the ovals, no darkening appeared there. Although this outcome was not the motivation for printing Scantegrity ballots with invisible ink, it appears evident that invisible ink yields a superior method for marking optical scan ballots. We supplied pointed “bullet”-style special pens to facilitate writing down the confirmation numbers. Wider “chisel”-style special pens, however, seem to work better for marking ovals.

Figure 4.16 shows correlations between survey responses on age and ease of use, and between understanding of Scantegrity and overall confidence in the system. As expected, overall, older voters found Scantegrity harder to use than did younger voters. Interestingly, most voters still had high confidence in Scantegrity, even if they felt they understood the system poorly. This finding runs contrary to a widely asserted notion that voters will not accept a system that they do not understand.

4.4 Recommended Changes and Lessons Learned

To simplify and streamline the process, we recommended the following:

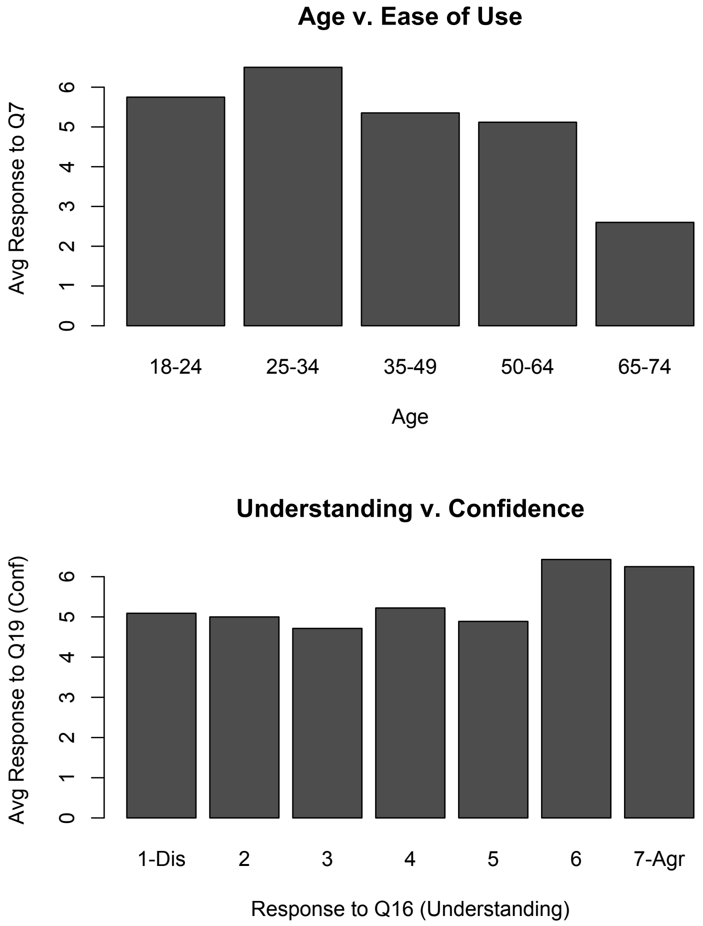


Figure 4.16: Correlation between age and overall ease of use, and between understanding and overall confidence in system. Voters under 65 years old found Scantegrity easier to use. Voters who felt they understood the system very well had slightly higher confidence in the system, yet even those who felt they had a poor understanding of the system had a moderately high confidence in the system. Pearson correlation coefficients: age vs. ease of use: -0.20, understanding vs. confidence: 0.28.

1. Eliminate the locked clipboard.

2. Eliminate redundant instructions. Do not provide instructions for optional steps at the beginning of the process.

3. Use high-quality, fast, robust scanners—preferably of the type that automatically drops the ballot into the ballot box when the voter signals to cast the ballot. The scanner should accept ballots inserted in any orientation.

4. Add a printer to the scanner to provide a digitally-signed receipt with the confirmation numbers. Great care must be taken to ensure that this printer does not violate ballot privacy.

5. Eliminate the tear-off chit. Instead, provide a separate sheet of paper to any voter who wishes to write down confirmation numbers or other ballot information by hand. Even with a receipt printer, voters should have the option of writing down confirmation numbers by hand.

6. Print confirmation numbers with a reduced character set to avoid easily confused letters.

7. Use "chisel"-style special pens for ease of marking ovals, selecting a small enough chisel width to permit writing down confirmation numbers and write-in candidates

8. Thoroughly analyze and test the voting process with many diverse voters.

The mock election demonstrated that Scantegrity can be effectively used in elections and is well accepted by voters, though the flow of people through the voting process must be greatly improved. The Scantegrity implementation, procedures, voter instructions, and equipment used in this election needed to be simplified and streamlined.

Chapter 5

Municipal Election at Takoma Park

On November 3, 2009, voters in Takoma Park, Maryland, cast ballots for the mayor and city council members using the Scantegrity II voting system. The election, with 1,728 voters from six wards, involved paper ballots with invisible-ink confirmation numbers, instant-runoff voting with write-ins, early and absentee (mail-in) voting, dual-language ballots, provisional ballots, privacy sleeves, any-which-way scanning with parallel conventional desktop scanners, E2E verifiability based on optional web-based voter verification of votes cast, a full hand recount, thresholded authorities, three independent outside auditors, fully-disclosed software, and exit surveys for voters and poll workers.

This municipal election marked the first time that anyone could verify that the votes were counted correctly in a secret ballot election for public office without having to be present for the entire proceedings. This chapter is a case study of the Takoma Park election, describing what was done—from the time our research team was approached by the Takoma Park board of elections in February 2008 to the last cryptographic election audit in December 2009—and what was learned. The focus of this chapter is the engineering process of bringing a new cryptographic approach to solve a complex practical problem involving technology, procedures, and laws.

All the software used in the election—for ballot authoring, printing, scanning and tally—was published well in advance of the election as commented, buildable source code, which may be a first in its own right. Moreover, commercial off-the-shelf scanners were adapted to receive ballots in privacy sleeves from voters, making the overall system relatively inexpensive.

Despite some glitches, the use of Scantegrity II was a success, demonstrating that E2E cryptographic voting systems can be effectively used and accepted by the general public. We found that the amount of extra work needed by officials to use Scantegrity II while administering an election is acceptable given the promise of improved voter satisfaction and indisputability of the outcome. Discussions are ongoing with the Takoma Park board of elections regarding continued use of the system.

The next section describes in more detail the setting for the election and gives details about Takoma Park and their specific election requirements. Section 5.2.1 provides an overview of the implementation changes from the mock election of the voting system for the November 3, 2009, Takoma Park municipal election, including the scanner software, cryptographic back end, and random number generation routines.

Section 5.3 gives a chronological presentation and timeline of the steps taken to run the November election, including the outcome of the voter verification and audits. It also gives the results of the election with some performance and integrity metrics. Section 5.5 discusses the high-level lessons learned from this election.

5.1 The Setting

For several reasons, the implementation of a new voting system in a jurisdiction is a difficult task. Most voting system users—*i.e.,* the voters—are untrained and elections happen infrequently. Voter privacy requirements preclude the usual sorts of feedback and auditing methods common in other applications, such as banking. Also, government regulations and pre-existing norms in the conduct of elections are difficult to change. These issues can pose significant challenges when deploying the new voting system, and it is therefore useful to understand the setting in which the election took place.

5.1.1 About Takoma Park

The city of Takoma Park is located in Montgomery County, Maryland; shares a city line with Washington, D.C; and is governed by a mayor and a six-member city council. The city has about 17,000 residents[1] and almost 11,000 registered voters [138, pg. 10]. A seven-member board of elections conducts local elections in collaboration with the city clerk. In the past, the city has used hand counts and optical scan voting as well as DREs for state elections.

The Montgomery County U.S. Census update data of 2005 provides some demographic information about the city. Median household income in 2004 was $48,675. The percentage of households with computers was 87.4%, and about 32% of Takoma Park residents above the age of 25 had a graduate, professional or doctoral degree. It is an ethnically diverse city: 45.8% of its residents identify their race as "White," 36.3% as "Black," 9.7% as "Asian or Pacific Islander" and 8.2% as "Other" (individuals of Hispanic origin form the major component of this category). Furthermore, 44.4% of its households have a foreign-born

[1] See http://www.takomaparkmd.gov/about.html

head of household or spouse, and 44.8% of residents above the age of five spoke a language other than English at home.

5.1.2 Agreement with the City

As with any municipal government in the U.S., Takoma Park is allowed to choose its own voting system for city elections. For county, state, and federal elections, it is constrained by county, state, and federal election laws.

Takoma Park and our research team signed a memorandum of understanding (MOU). Part of that agreement included the mock election, which is described in Chapter 4. After the mock election, the Takoma Park board would determine what changes would be necessary and make a final decision on whether to recommend use of Scantegrity for the municipal election; the board did make this recommendation. The city council had final say on the board's recommendations; it adopted all recommendations.

During the election, we agreed to provide equipment, software, training assistance, and technical support. The City of Takoma Park agreed to provide election-related information on the municipality, election workers, consumable materials, and perform or provide all other election duties or materials not provided by us. No goods or funds were exchanged.

According to the MOU, if approved by the city council, the election was to be conducted in compliance with all applicable laws and policies of the city. This included supporting requirements of Instant Runoff Voting, early voting, multiple languages on the ballot (specifically Spanish), absentee voting, multiple ballot styles, and other rules as defined by the City of Takoma Park Municipal Charter.[2]

We also agreed to pursue an accessible ballot-marking device for the election, but we were later relieved of satisfying this requirement. Unfortunately, Scantegrity is not yet

[2] http://www.takomaparkmd.gov/code/pdf/charter.pdf

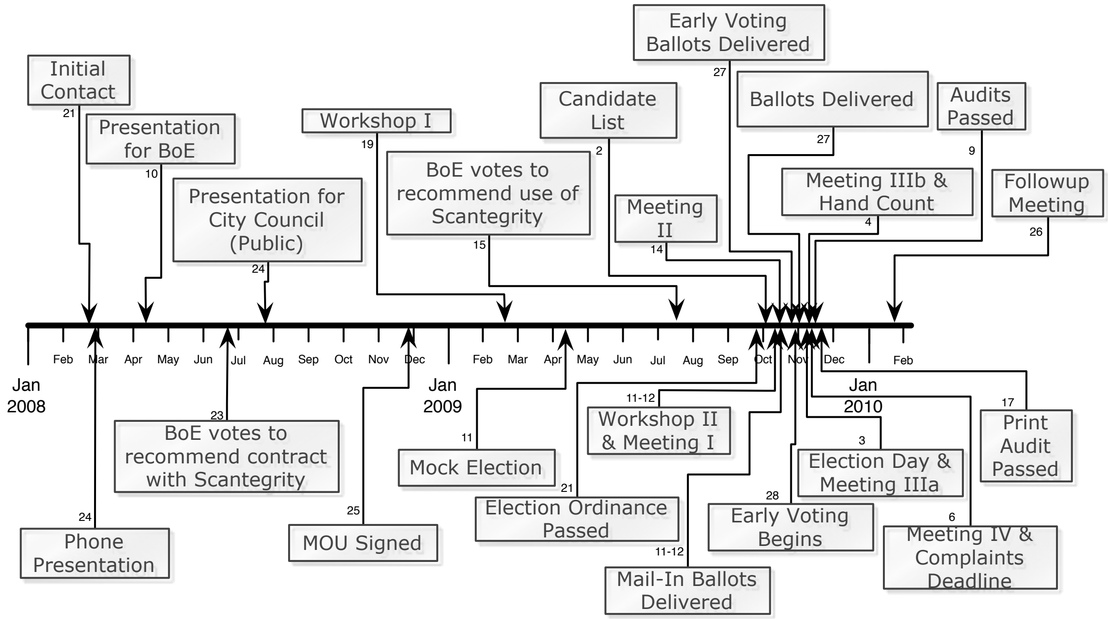


Figure 5.1: Takoma Park 2009: Timeline Illustration. Each box represents an event, and the numbers next to each box are the day(s) the event took place during the indicated month.

fitted with a voter interface for those with visual or motor disabilities, and accessible user interfaces were not used in Takoma Park's previous optical scan elections.

5.1.3 Election Timeline

As mentioned in Chapter 4, our research team was first approached by the Takoma Park board of elections in late February 2008. We were invited to give a presentation on the system, and, after two presentations and consideration of other voting systems, the board voted to recommend a memorandum of understanding (MOU) with Scantegrity in June 2008. (see Figure 5.1). Following a public presentation to the city council in July 2008, the MOU was signed in late November 2008, about 9 months after the initial contact. See section 5.1.2 for details on the agreement.

Our team held an open workshop in February 2009 to discuss the use of Scantegrity in both a mock and real election. This workshop was held at the Takoma Park Community Center and was attended by board of election members, the city clerk, current members (and a retired member) of the Montgomery County Board of Elections, and a representative each from the Pew Trust and FairVote. Following the mock election in April 2009, discussed in Chapter 4, we proposed a redesigned system that took into consideration feedback from voters and poll workers (through surveys) and the board of elections. The board voted to recommend use of the redesigned system in July 2009; this was made official in the city election ordinance in September 2009.[3] Beginning around June 2009, a meeting with representatives from our research team was on the agenda of most monthly board of election meetings. Additionally, our team members met many times with the city clerk and the chair of the board of elections to plan for the election.

The final list of candidates was available approximately a month before the election, on October 2. The Scantegrity *meetings* initializing the data and ballots were held in October (see Section 5.3), as was a final workshop to test the system. Absentee ballots were sent out by the City Clerk in the middle of October. Our team delivered ballots to the city clerk in late October, and early voting began almost a week before the election on October 28. Poll worker training sessions were held by the city on October 28 and 31, and polling was held on November 3, 2009, from 7 am to 8 pm. The final Scantegrity audits were completed on 17 December 2010; all auditors were of the opinion that the election outcomes were correct (for details, see Section 5.3).

[3] See http://www.takomaparkmd.gov/clerk/agenda/items/2009/090809-3.pdf, section 2-D, page 2.

5.1.4 Instant Runoff Voting (IRV)

Takoma Park has used IRV in municipal city elections since 2006, and IRV was the only unique requirement for this election. Other requirements such as early voting, multiple languages on the ballot, absentee voting, multiple ballot styles, and other rules—while there are many variations between jurisdictions—are reasonably standard features seen in many elections throughout the United States at the time of this writing.

IRV is a ranked choice system in which each voter assigns each candidate a rank according to her preferences. Thus, the preferred candidate should be ranked first, the least preferred candidate last, and all others in between. In Takoma Park, voters do not have to choose a rank for each candidate. However, counting stops for a ballot if more than one rank is skipped or two candidates share the same ranking on the ballot.

When counting, the topmost choice on each ballot is counted as one vote for the chosen candidate. If a candidate receives a majority, defined as greater than 50% of the valid votes in a round, he is declared the winner. If no candidate receives a majority, the candidate with the fewest votes is eliminated, and counting proceeds for another round. In each round, ballots for eliminated candidates are reviewed to find highest choice of advancing uneliminated candidates. If an advancing uneliminated candidate is found, that ballot is counted toward that candidate, and if no advancing candidates are found the ballot is considered exhausted. If no candidate receives a majority, the last surviving candidate is declared the winner.[4]

Ties can occur when eliminating candidates. If the total of votes for all tied candidates is less than the next lowest candidate, all the tied candidates are dropped. Otherwise, each preceding round is evaluated, and if a candidate with less votes than the others is found, that

[4] For the exact laws used by Takoma Park, see page 22 of http://www.takomaparkmd.gov/code/pdf/charter.pdf. Section (f), concerning eliminating multiple candidates, was used in our implementation for tie-breaking only.

candidate is eliminated first. If the tie cannot be broken this way, the tie is resolved by lot (e.g., drawing straws). In our software, every possible elimination is evaluated in turn.

5.2 Implementation Changes from the Mock Election

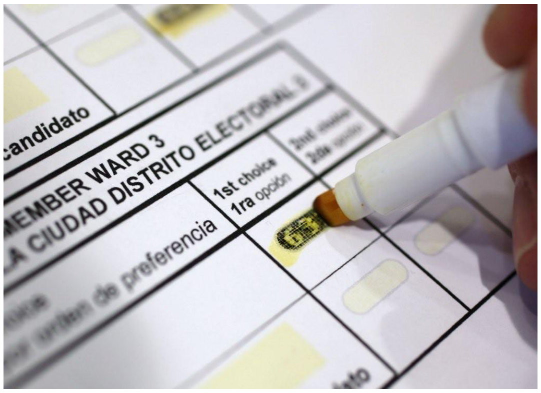
As per the timeline in Section 5.1.3, we had less than seven months after the Mock election (and less than four after the system was approved) to make modifications to the election system. This presents a practical problem: We could not make all the changes we had recommended from the Mock election, and we had to decide which changes we could make in time for the election. In this section, we discuss what changed from the implementation used during the Mock election.

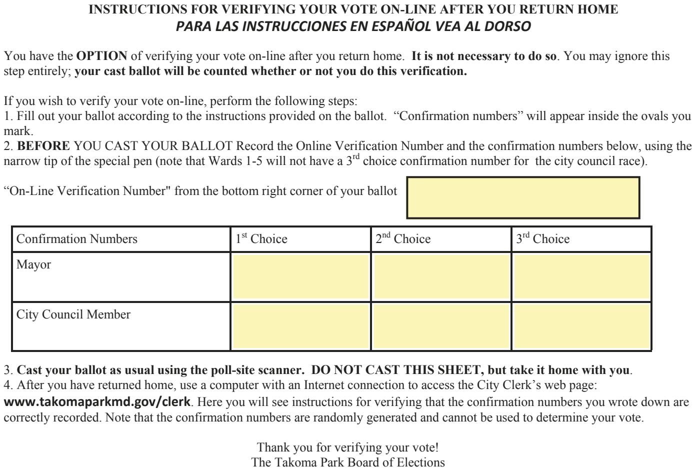
5.2.1 Changes to the Ballot and Voting Procedures

The Takoma Park board of elections and our team made several ballot and voting procedure changes. We designed these changes in order to streamline the voting process.

The Scantegrity ballot now looks more similar to a conventional optical scan ballot (see Figure 5.3 for a sample ballot used in the election). It contains a list of the choices and bubbles beside each choice. Marking a bubble reveals a random 3-digit confirmation number. The detachable area was removed and is now a separate verification card (see Figure 5.2).

At a high level, the voter experience has changed to be as follows: First, a voter checks in at the polling place and receives a Scantegrity ballot (See Figure 5.3) with a privacy sleeve instead of the locked clipboard. The privacy sleeve is used to cover the ballot and keep its contents private. Inside the voting booth are the decoder pen and a stack of blank voter verification cards. The voter uses the decoder pen to mark the ballot. As on a conventional optical scan ballot, she fills in the bubble next to each of her selections.





INSTRUCTIONS FOR VERIFYING YOUR VOTE ON-LINE AFTER YOU RETURN HOME

PARA LAS INSTRUCCIONES EN ESPAÑOL VEA AL DORSO

You have the **OPTION** of verifying your vote on-line after you return home. **It is not necessary to do so**. You may ignore this step entirely; **your cast ballot will be counted whether or not you do this verification.**

If you wish to verify your vote on-line, perform the following steps:
1. Fill out your ballot according to the instructions provided on the ballot. "Confirmation numbers" will appear inside the ovals you mark.
2. **BEFORE** YOU CAST YOUR BALLOT Record the Online Verification Number and the confirmation numbers below, using the narrow tip of the special pen (note that Wards 1-5 will not have a 3rd choice confirmation number for the city council race).

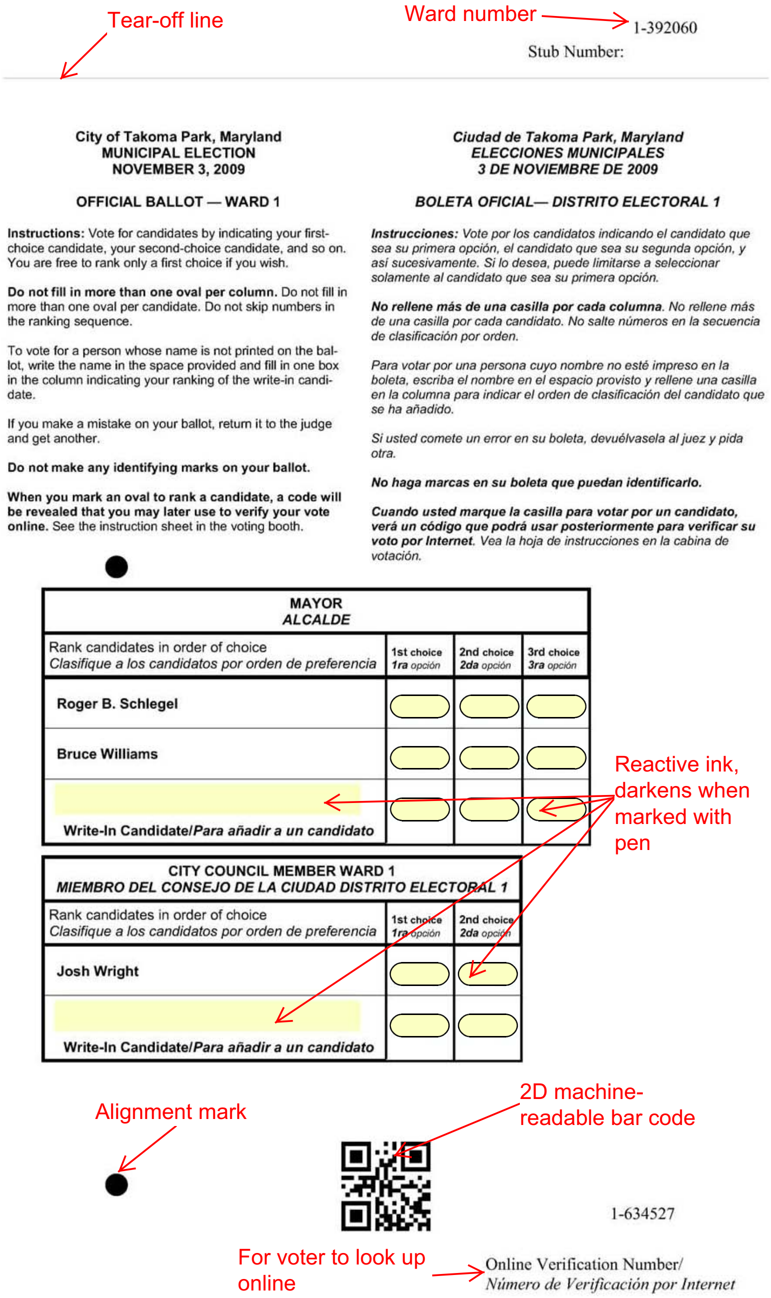
"On-Line Verification Number" from the bottom right corner of your ballot

Confirmation Numbers	1st Choice	2nd Choice	3rd Choice
Mayor			
City Council Member			

3. **Cast your ballot as usual using the poll-site scanner. DO NOT CAST THIS SHEET, but take it home with you**.
4. After you have returned home, use a computer with an Internet connection to access the City Clerk's web page: **www.takomaparkmd.gov/clerk**. Here you will see instructions for verifying that the confirmation numbers you wrote down are correctly recorded. Note that the confirmation numbers are randomly generated and cannot be used to determine your vote.

Thank you for verifying your vote!
The Takoma Park Board of Elections

Figure 5.2: Left: a portion of a marked ballot for Ward 3, showing exposed digits of the confirmation number when the decoding ink reacts with the reactive ink in the oval. Notice the chisel tip of the pen. Picture by A. Rivest. Right: The verification card for writing down confirmation numbers. Actual size is 8.5 by 11 inches; Spanish instructions were located on the back. Because the marking areas are printed on with reactive ink, the same pen can be used to mark the ballot and to note confirmation numbers. Two-sided pens with chisel and regular tips were used for the election.


Tear-off line
Ward number
Reactive ink, darkens when marked with pen
Alignment mark
2D machine-readable bar code
For voter to look up online

1-392060

Stub Number:

City of Takoma Park, Maryland
MUNICIPAL ELECTION
NOVEMBER 3, 2009

OFFICIAL BALLOT — WARD 1

Instructions: Vote for candidates by indicating your first-choice candidate, your second-choice candidate, and so on. You are free to rank only a first choice if you wish.

Do not fill in more than one oval per column. Do not fill in more than one oval per candidate. Do not skip numbers in the ranking sequence.

To vote for a person whose name is not printed on the ballot, write the name in the space provided and fill in one box in the column indicating your ranking of the write-in candidate.

If you make a mistake on your ballot, return it to the judge and get another.

Do not make any identifying marks on your ballot.

When you mark an oval to rank a candidate, a code will be revealed that you may later use to verify your vote online. See the instruction sheet in the voting booth.

Ciudad de Takoma Park, Maryland
ELECCIONES MUNICIPALES
3 DE NOVIEMBRE DE 2009

BOLETA OFICIAL— DISTRITO ELECTORAL 1

Instrucciones: *Vote por los candidatos indicando el candidato que sea su primera opción, el candidato que sea su segunda opción, y así sucesivamente. Si lo desea, puede limitarse a seleccionar solamente al candidato que sea su primera opción.*

No rellene más de una casilla por cada columna*. No rellene más de una casilla por cada candidato. No salte números en la secuencia de clasificación por orden.*

Para votar por una persona cuyo nombre no esté impreso en la boleta, escriba el nombre en el espacio provisto y rellene una casilla en la columna para indicar el orden de clasificación del candidato que se ha añadido.

Si usted comete un error en su boleta, devuélvasela al juez y pida otra.

No haga marcas en su boleta que puedan identificarlo.

Cuando usted marque la casilla para votar por un candidato, verá un código que podrá usar posteriormente para verificar su voto por Internet*. Vea la hoja de instrucciones en la cabina de votación.*

MAYOR ***ALCALDE***			
Rank candidates in order of choice *Clasifique a los candidatos por orden de preferencia*	**1st choice** ***1ra** opción*	**2nd choice** ***2da** opción*	**3rd choice** ***3ra** opción*
Roger B. Schlegel	○	○	○
Bruce Williams	○	○	○
Write-In Candidate/*Para añadir a un candidato*	○	○	○

CITY COUNCIL MEMBER WARD 1 ***MIEMBRO DEL CONSEJO DE LA CIUDAD DISTRITO ELECTORAL 1***		
Rank candidates in order of choice *Clasifique a los candidatos por orden de preferencia*	**1st choice** ***1ra** opción*	**2nd choice** ***2da** opción*
Josh Wright	○	○
Write-In Candidate/*Para añadir a un candidato*	○	○

1-634527

Online Verification Number/
Número de Verificación por Internet

Figure 5.3: An unmarked Takoma Park 2009 ballot for Ward 1 showing instructions in Spanish and English, the options, the circular alignment marks, the 2D barcode, the ballot serial number (on the stub, meant for poll workers to keep track of the number of ballots used), and the online verification number (for voters to check their codes). The true ballot was printed on legal size paper and was hence larger than shown.

If the voter wishes to verify her vote later on the election website, she can copy her ballot verification ID and revealed confirmation numbers onto a voter verification card. She keeps the verification card for future reference. She then takes her ballot to the scanning station and feeds the ballot into an optical scanner, which reads the ballot ID and the marked bubbles. The scanner no longer detects over or undervoted ballots and accepts any paper scanned into it directly into the ballot box.

If a voter makes a mistake, she can ask a poll worker to replace her ballot with a new one. The first ballot is marked "spoiled" and its ballot ID is added to the list of spoiled ballot IDs maintained by the election judges. Voters can still audit ballots but this is not explained; instead, outside auditors perform this step.

The voter can verify her vote on the election website by checking that her revealed confirmation numbers and ballot ID have been posted correctly. If she finds any discrepancy, the voter can file a complaint through the website, within a complaint period. When filing a complaint, the voter must provide the confirmation numbers that were revealed on her ballot as evidence of the validity of the complaint.

Because there are no longer chit serial numbers and there is no rejection at the scanner, it is not possible for voters to prove if ballots have been overvoted to invalidate them or if undervoted ballots have been used for voting. Also, ballot secrecy is slightly worse, because the verification serial number is not removed from the ballot after voting.

We made sure that Takoma Park made these decisions knowing the effect they would have on the security properties of the system. Stolen ballots and chain voting attacks were not considered a threat by Takoma Park because the voting area has only one entrance and exit (election judges were posted on both, and were instructed not to allow voters to leave unless they witnessed the voter scanning a ballot or another judge escorted them out.) To deal with undervotes being switched to votes, officials told voters to overvote their ballot and record both numbers. To deal with regular votes being switched to overvotes, officials relied on chain of custody and posted results from the scanners as elections closed, reducing the window of opportunity.

5.2.2 Confirmation Numbers

Recall that confirmation numbers are unique within each contest on each ballot, and are generated independently and uniformly pseudorandomly. The confirmation number corresponding to any given choice on any given ballot is hidden and unknown to any voter until the voter marks the bubble for that choice.

Confirmation codes were changed from an alphanumeric character set to a numeric-only character set. We proposed this change to Takoma Park to reduce the likelihood of confusion between letters we saw during the mock election. To maintain the level of security with the reduced character set, we printed three characters instead of two.

5.2.3 Website Bulletin Board

After the election, any interested party can audit the election by using software to check the correctness of the data and final tally on the election website. Beyond communicating the election outcome itself, the role of the election website is to serve as a "bulletin board" (BB) to broadcast the cryptographic audit data set (*i.e.,* cryptographic commitments, responses to audit challenges). In addition, voters can use this website to check their receipts and file a dispute if the receipt is misreported. We only used part of the mock election implementation of the bulletin board.

Originally, our plan was to have Takoma Park host the website, but officials chose a hybrid approach in which they agreed to host election information and results. Their website would link to our server to provide a receipt checking tool and audit data. After the election, officials would provide us with a copy of the public data files to publish. This decision caused a number of changes to our approach.

We decided only to use the receipt checking code from the implementation and to make downloading more convenient for auditors, post all election data on our publicly available subversion repository.[5] Additionally, our auditors agreed to mirror the data.

A primary security requirement for the Scantegrity BB is to provide authenticated broadcast communication from election officials to the public. We met this requirement with digital signatures. A team member (Carback) created signed copies of each file with gnupg[6] using his public key from May 28, 2009.

Without authenticated communication, it would be impossible to prove whether different results were provided to different people. Our specific approach to the website requires observers to verify signatures and check with each other if they receive identical copies of the data (and verify the consistency of the signatures over time). Our auditors, Adida and Zagorski, performed these actions, but we do not know the extent of this communication otherwise. As usual with our approach to Scantegrity, we enable **detection** of errors (genuine or malicious).

There are several potential threats to the bulletin board model. At a high level, threats pertain primarily to misreporting of results or to voter identification. With regard to results reporting, an adversary may attempt to misreport results by substituting actual election data with false data. In the event that all parties verify signatures of information they receive and check consistency with the signed files, incorrect confirmation numbers on the bulletin board would be detected by voters, and incorrect computation of the tally by anyone

[5] See http://scantegrity.org/svn/data/takoma-nov3-2009

[6] See http://www.gnupg.org/

checking the tally computation audit. If the voter checking confirmation numbers does not check consistency with the rest of the bulletin board (by, for example, downloading the bulletin board data, checking all the signatures, and checking that his or her confirmation number is also correctly noted in the entire bulletin board data) he or she may be deceived into believing their ballot was accurately recorded and counted. Similarly, if the various signatures are not cross-checked across individuals or observed over time, an adversary may replace the confirmation numbers after they have been checked, or send different ones to voters and to auditors. An adversary may also attempt an identification attack in which the objective is to link voter identities with receipt data (e.g., by recording IP addresses of voters who check their receipts).

If the actual election data are incorrect, results substitution could serve as an attempt to cover up fraud or error. If the substitution takes place on the election website, the discrepancy will be detectable during the cryptographic tally audit of that data set. An adversary attempting to do this may, as a strategy, attempt to install malicious code on a voter's computer to intercept and instead display the substituted data. In Scantegrity, no private computation takes place on the client device, and because the voter can use any web-connected device to perform this step, this strategy appears to be unscalable.

5.2.4 Auditing

At the polling place on the day of the election, any interested party could choose to audit the printing of the ballots. A print audit consists of marking all of the bubbles on a ballot, and then either making a photocopy of the fully-marked ballot or copying down all of the revealed confirmation numbers. The ballot ID is recorded by an election judge as audited. After the election, anyone can check that all of the confirmation numbers on the audited ballot—and their correspondence with ballot choices—are posted correctly on the election website. One of the independent auditors, Coney, participated in this process.

Random numbers are needed to generate challenges for the various auditing steps (print audit, randomized partial checking). These numbers should be unpredictable in advance to an adversary. They should also be verifiable after the fact as having come from a "truly random" source that is not manipulable by an adversary.

We chose to use the closing prices of the stocks in the Dow Jones Industrial Average as our verifiable and unpredictable source to seed the pseudorandom number generator (the use of stock prices for this purpose was first described in [45]). These prices are sufficiently unpredictable for our purposes, yet verifiable after the fact. However, it was later discovered that post-closing "adjustments" can sometimes be made to the closing prices, which can make these prices less than ideal for our purposes in terms of verifiability.

5.2.5 Scanner Software

Our modified optical scan system uses EeePC 900 netbooks and Fujitsu 6140 scanners. Instead of the software calling the scanning program directly, we use a polling method. A bash shell script calls the SANE scanimage program[7] and polls a directory on the filesystem to acquire ballot images. Once an image is acquired an algorithm modified for speed uses circular alignment marks to adjust the image, reads the barcode using the ZXing QRCode Library,[8] and uses a simple threshold algorithm to determine if a mark is made on the ballot.

We removed the GUI and the reject option from the scanner. We could not make it work mechanically; elections officials determined that to be a significant potential source of slowdowns during the mock election. We also removed the scanner tally code and moved it

[7] See `http://www.sane-project.org/`

[8] See `http://code.google.com/p/zxing/`

to the tabulator system described in Section 5.2.6. Multiple scanners can now be used at each polling site, and memory cards from each scanner are added to the tabulator software.

We added multiple ballot style support to the scanner. Individual races on each ballot are identified by ward information in the barcode, which is nonsequential and randomly generated. The ballot ID in the barcode and the web verification numbers on each ballot are different numbers, and the association between each number type is protected by the back end system. Write-in candidate areas, if that candidate is selected by the voter, are stored as clipped raw images with the ballot scan results. Ballot scan results are stored in a random location in a memory-mapped file.

The current implementation of the scanning software still does not protect data in transit to the back end, which poses a risk for denial of service. Checking the correctness of the scanner is performed through the Scantegrity audit. The data produced by the scanner does not compromise voter privacy, but—assuming an attacker could intercept scanner data—voter privacy could be compromised at the scanner through unique write-in candidates on the ballot, through a compromised scanner, by bugs in the implementation, or by relying on the voter to make readable copies of the barcode to get a ballot ID.

5.2.6 Tabulator and Write-In Software

At the request of Takoma Park we created an additional piece of software, the Election Resolution Manager (ERM), that allows election judges to determine manually which candidate receives each write-in vote. The other responsibility of the ERM is to act as part of the back end. It collates data from each scanner and prepares the input files for the back end, acting as a tabulator.

To resolve write-ins with this software, the user cycles through each image and types in the name of the intended candidate or selects the name from a list of previously-identified candidates comprised of the original candidates and any previously-typed candidate names. Because the user is not shown the whole ballot, he does not know what the other selections are on that ballot or what rank the write-in was given. We call this process *resolving* a vote because the original vote is changed from the generic write-in candidate to the candidate who was intended by the voter. The ERM produces a PDF of each image, the candidate selection for that image, and a unique number to identify the selection.

Scantegrity handles write-in candidates just like other optical scan systems by treating the write-in position as a candidate. Therefore, the back end does not know how each write-in position was resolved, and two results records are created: one with write-in resolution provided by the ERM, and one without write-in resolution provided by the back end.

To check the additional record generated by the ERM, an observer reduces the resolved results record and verifies that the set of resolved ballots is the same as the set of unresolved ballots. To audit that the judges chose the correct candidates for each write-in, the observer refers to the PDF generated during write-in resolution. The PDF allows the observer to reference each resolved ballot entry in the resolved results file and verify that the image was properly transcribed.

One caveat of this approach is that if a write-in candidate wins, a malicious authority could modify these images to change results, but could not deny that the write-in position had received a winning number of votes. This situation would require additional procedures to verify the write-ins (e.g., a hand count or careful audit of the transcriptions by each judge).

5.3 The Election

In this section, we describe the election events chronologically.

5.3.1 Preparations

Preparations for the election include designating auditors, running the first two back end meetings, and creating the ballot.

Independent Auditors

The board of elections requested cryptographers Ben Adida (Center for Research on Computation and Society, Harvard University) and Filip Zagórski (Institute of Mathematics and Computer Science, Wroclaw University of Technology, Poland) to perform independent audits of the digital data published by Scantegrity in general, and of the tally computation in particular. Adida[9] and Zagórski[10] maintained websites describing the audits and the results of the audits, and Adida also blogged the audit.[11] Before the election, Adida pointed out several instances when the Scantegrity information was insufficient; Scantegrity documentation was updated as a result.

The board of elections also requested Lillie Coney (Associate Director, Electronic Privacy Information Center and Public Policy Coordinator for the National Committee for Voting Integrity (NCVI)) to perform print audits on election day. Coney chose ballots at random through the day, exposed the confirmation numbers for all options on the ballot, and kept these with her until after the end of the complaint period, when Scantegrity opened commitments to all unvoted and unspoiled ballots (and hence to all ballots she had audited). Coney then checked that the correspondence between codes and confirmation numbers on her ballots matched those on the website.

Both tasks, of print audits and digital data audits, can be performed by voters. Digital data audits can also be performed by any observers. In future elections, when the general

[9] See http://sites.google.com/site/takomapark2009audit/

[10] See http://zagorski.im.pwr.wroc.pl/scantegrity/

[11] See http://benlog.com/articles/category/takoma-park-2009/

population and Takoma Park voters are more familiar with end-to-end elections, we expect that voters (particularly candidate representatives) will perform such audits.

Meeting 1

Four election officials (the city clerk, the chair, vice chair and a member of the board of elections: Jessie Carpenter, Anne Sergeant, Barrie Hofmann and Jane Johnson, respectively) were established as election trustees at *Meeting 1*, held on October 12, 2009.

It was explained to the trustees that, through their passwords, they would generate the confirmation numbers and share the secret used to tally election results. Further, it was explained that, without more than a threshold of passwords, the election could not be tallied by Scantegrity, and that if a threshold number of passwords was not accessible (e.g., if they were forgotten or trustees were unavailable due to sickness) the only available counts would be manual counts. A threshold of two trustees was determined based on anticipated availability of the officials, and it was explained that two trustees could collude to determine the correspondence between confirmation numbers and codes; hence, each trustee should keep her password secret.

The trustees generated commitments to the decryption paths for each of 5,000 ballots per ward (for six wards). Scantegrity published the commitments on October 13, 2009 at 12:13 am.

Meeting 2

In *Meeting 2*, held on October 14, 2009, trustees used our sourcecode to respond to challenges generated using stock market data at closing on October 14. Half of the ballot decryption paths committed to in *Meeting 1* were opened. Additionally, trustees constructed ballots (associations between candidates and confirmation numbers) at this meeting and generated commitments to them. Scantegrity published the stock market data, the challenges, and the responses.

Ballot Design

The 2009 ballot was based on ballots used for the 2007 election. We chose to modify (as little as possible) a design used successfully in a past election rather than use the ballot we had designed for the mock election because the previously-used ballot design would be familiar to voters. The ballot was required to contain instructions in both English and Spanish: marking instructions, instructions for write-ins, instructions for IRV, and any Scantegrity-related instructions (see Figure 5.3).

Printing Ballots

We began printing with six printers; however, they proved unreliable. It was our expectation that using large amounts of commodity hardware would scale, but it did not. We did not anticipate the number of failure modes we experienced and our printing process was delayed by approximately 1.5 days. The failures include:

1. Clogged inkjet heads caused the confirmation numbers to become readable when they should have been hidden.

2. Faulty dispenser cartridges prevented printing or spilled out onto the pages. Many cartridges became faulty during the printing process, and we suspect properties of the reactive and dummy inks caused the problems.

3. Misaligned inkjet heads caused confirmation numbers to be unreadable.

4. At least one printer appeared to have been broken out of the box.

5. Many of the printers would stop working intermittently after a print job had finished. We suspect the cartridge emulation chips we used from the refillable ink cartridge system may have interacted poorly with the firmware.

6. A component failed in one of the printers during the printing. It released fumes, and we waited outside for the fumes to dissipate.

Prior to these problems, our tests when printing ballots were successful. The only failure mode we were aware of was when printers were turned off for a prolonged period of time: they developed clogged print heads that could not be cleaned.

We detected most issues by printing a test page every 25 pages. If the test page was bad, the previous ballot stack was destroyed, and we worked to resolve the issue before printing again.

To fix the clogging issue, we used the head-cleaning function on the printers or replaced the cartridges. We also replaced cartridges when they were faulty. Misalignment was fixed using the align-heads function on the printers. We bought additional printers to replace those that were not working.

Ballot Delivery

Mail-in (absentee) ballots were delivered to the city clerk on October 16. Early in-person ballots were delivered on October 27 for early voting on October 28, and all other ballots on October 30.

Absentee ballots were identical to in-person voting ballots except they did not contain online verification numbers and voters were not given any instructions on checking confirmation numbers online. They were returned by mail in double envelopes and scanned with the early votes. Confirmation numbers for these ballots were, however, made available online after scanning, so that there was no distinction in published data between absentee and in-person voted ballots.

The board decided to issue ballots without confirmation numbers due to the small number of anticipated absentee votes and the costs associated with mailing ballots with special pens. Mailing ballots with confirmation numbers would allow verification of confirmation numbers, but would open up new attacks: the possibility of false charges of election fraud by adversaries who might expose confirmation codes and reprint ballots or use expensive equipment to attempt to determine the invisible codes. Strong verification for absentee ballots is an ongoing research subject within the Scantegrity team.

Early in-person voters used Scantegrity ballots with all Scantegrity functionality; however, the early votes were scanned after the polls closed on election day rather than by voters themselves. Voters were, however, provided verification cards and could check confirmation numbers for these ballots online.

Poll Worker Training

Several training sessions were held in the weeks prior to the election. Manuals from the previous election were updated and a companion guide was created with Scantegrity-specific instructions. Election judges were given these two manuals, and a member from our team demonstrated the voting process at one session.

Voter Education

Voter education for this election focused on online verification. Articles in the city newspaper before the real election indicated that voters could check confirmation numbers online; this was also announced on the city's election website.[12]

[12] See http://www.takomaparkmd.gov/clerk/election/2009/

Because it was not clear whether a sufficiently large number of voters would attend or benefit from a voter instruction session, none was held before the election. As such, one-on-one voter education on the Scantegrity ballot and Scantegrity voting system was minimal. There is much for voters to process about E2E voting systems: the ability to verify vote encryptions on the website, the ability to check the computations of the encrypted votes (the tally audit), and the ability to challenge encrypted votes (the print audit). It is reasonable to expect that it will take a few elections to educate a community of voters about all the features. Now that voters are aware of the properties of Scantegrity, it is expected that there will be more interest in learning more about the system, and that our team and Takoma Park will hold voter education sessions and instructional and/or sample ballot mailings if the system is used again.

Scanner Setup

We attempted to minimize, not prevent,[13] the potential for using the wrong software by installing our software on Ubuntu Linux on SD flash cards, setting the "read-only" switch on each card, and setting up the software to read and write to USB sticks. We fingerprinted the first card after testing with the sha1sum utility and cloned it to a second card for the other netbook. Each netbook was set to boot from the card and the BIOS configuration was locked with a password.

Both flash cards were checked with the sha1sum utility then placed into the netbook which was placed into a lockbox and delivered to Takoma Park. The USB sticks were initialized with scanner configuration files. We uniquely identified each scanner by changing

[13] Scantegrity would detect manipulation at the scanner. A better solution would use trusted hardware technology (e.g. a Trusted Platform Module [62]).

the ScannerID field in the configuration files, then placed the corresponding USB sticks (three for each netbook) into the lockbox.

Upon delivery of the scanners the day before the election, we gave election officials the lockbox keys and showed them how to open the lockboxes. We confirmed with election officials the contents of each box and the officials verified, with our assistance, that the USB memory sticks did not contain any ballot data by looking at the configuration file and making sure the ballot data file was blank.[14] To protect against virus infection on the sticks we set them to read-only for this procedure.

5.3.2 Election Day

On election day—November 3, 2009—polls were open from 7 am to 8 pm at a single polling location, the Takoma Park Community Center. Several members of our team were present for most of the day in case of technical difficulty. One member of our team was permitted in the polling room at most times as an observer, and a couple of team members were present in the vestibule giving out and collecting survey forms through most of the day. Lillie Coney of the Electronic Privacy Information Center, who performed a print audit at the request of the board of elections, was present in the polling room through a large part of the day.

Starting the Election

The scanner was the only equipment to set up, and it was a turnkey system. Election judges needed to plug in the USB sticks and power on the netbooks. The scanner was attached to a scanning apparatus, and cables were run into the lockbox that contained the netbook. When ready, the scanner beeped three times. After reading a ballot, the scanner beeped once.

[14] These were the only two files on the disk at this time. Additionally, election officials did not check fingerprints on the flash cards. Since no third party had reviewed the code or fingerprinted it, they relied on our chain of custody.

During shutdown, the scanner beeped another three times. If there were any failure modes, the scanner beeped continuously or notat all.

Election judges set up the check-in tables, pollbooks, and voting booths. The election started on time.

Voting

The election proceeded quite smoothly, with only a few small glitches. A team member was able to assist polling officials in fixing a problem with their pollbooks (not provided by Scantegrity). Voters had some initial problems with the use of the scanner and the privacy sleeve; some sought assistance from election judges, who also had difficulty. After an explanation to the election judges by the chair of the board of elections, the use of the scanner was considerably smoother. The privacy sleeve would not separate from a few ballots; one ballot was mangled considerably, but scanned fine. Seventeen scanned ballots had lines on them that prevented the scanner from reading votes, and one ballot had alignment marks manipulated such that it was also unreadable. Images of all unreadable scans are saved, so we were able to manually enter in these votes. Of the 17 ballots, many ballots had a line in the same location, which is consistent with the presence of a foreign substance on a ballot put into the scanner. These problems did not affect our ability to count the votes.

During the day, Coney chose about fifty ballots at random, uniformly distributed across wards, and exposed the confirmation numbers for all options for the ballots. A copy of each ballot was made for her to take with her; the copies were signed by the chair. Neither Coney nor our team had any interaction with voters.

Towards the end of the day, after the local NPR station carried clips from an interview with the chair of the board of elections and a voter, the polling station saw a large increase in the number of voters; the line took up much of the floor outside the polling room.

Some voters were curious about the verifiability properties of the system. Our team prepared to print more ballots, but this was not required. The number of printed ballots was almost twice the number of voted ballots.

Absentee and early voted ballots were scanned in after the closing of polls. Afterward, the scanners were shut down. The chief judge opened each lockbox, set all sticks to read only, removed two USB sticks (leaving the third with the scanning netbook), and locked the lockbox. Our team was given one stick for the ERM system. The other was kept by the city.

In *Meeting 3a*, trustees used Scantegrity code to generate results without provisional ballots at about 10 pm. The chair of the board of elections announced the results to those present at the polling place at the time (including candidates, their representatives, voters, and others); this was also broadcast live by the local TV station. Confirmation codes and the election day tally were posted on the Scantegrity website.

5.3.3 After the Election

The next day, around 2 pm, results—including verified provisional ballots—were published. Takoma Park representatives had announced a tally without provisional ballots the night before, and followed with the tally that included verified provisionals in accordance with standard Takoma Park procedures. The final *Meeting 3* results were published on November 4 just before midnight.

There were 10,934 registered voters and 1728 votes were cast (15.8%). The city-certified final tally for each contest is provided in Table 5.1. In each race, a majority was won after tallying the first round of voting.

Mayor	Votes
Roger B. Schlegel	664
Bruce Williams	1000
Write-ins	17

Ward	Councilor	Votes	Ward	Councilor	Votes
Ward 1	Josh Wright	434	Ward 4	Terry Seamens	196
	Write-ins	13		Eric Mendoza	12
Ward 2	Colleen Clay	236		Write-ins	2
	Write-ins	15	Ward 5	Reuben Snipper	71
Ward 3	Dan Robinson	397		Write-ins	10
	Write-ins	34	Ward 6	Navid Nasr	61
				Fred Schultz	138
				Write-ins	0

Table 5.1: City certified election results for the Mayor's race and each City Councilman's race.

Hand Count and Certification

Following a hand count performed by representatives from our team and Takoma Park, the chair of the board of elections certified the results of the hand count to the city council at 7 pm on November 5. The hand count and Scantegrity count differed because officials were able to better determine voter intent during the hand count. For example, in the mayoral race, the scanner count determined that 646 votes were cast for candidate Schlegel, 972 for Williams, 15 for various write-in candidates, and 90 were not cast. The certified hand count totals were 664 votes for Schlegel, 1,000 for Williams and 17 for write-in candidates. Thus 48 of a total of 1681 votes in this race would not have been counted by a scanner count alone. The discrepancy was caused by voters marking ballots outside of the designated marking areas. Such marks, while not read by the scanner by definition, are considered valid votes by Takoma Park law. Similarly, 8 of a total of 447 votes for Ward 1 council member, 8 of 251 for Ward 2, 16 of 431 for Ward 3, 10 of 210 for Ward 4, 2 of 81 for Ward 5 and 11 of 199 for Ward 6 were added to scanner vote totals after hand counting.

The hand count was the only audit of Scantegrity's electronic count that preceded certification. Scantegrity audits could not be held until all voters had been given a chance to complain about missing or manipulated confirmation numbers, and, unlike other jurisdictions, a Takoma Park election is typically certified the day after it is held. For a system and a paradigm (E2E voting) that had not been tested before in a governmental election, and that enforced greater accountability, it was particularly important to allow election officials to perform some audit prior to certification. The hand count was also an opportunity for our team to experience an important aspect of a regular election, and observe the differences between hand count and machine count results (e.g., the interpretation of voter intent). For future elections, we expect the audit to happen before certification.

Post-Election Audit

During *Meeting 4*, held on November 6 at 6 p.m., trustees used Scantegrity-written source-code to reveal all codes on voted ballots and reveal everything on all the ballots that were not spoiled or voted upon. Trustees also responded to pseudo-random challenges generated by stock market results at closing on November 6. Scantegrity published all data on November 7 around 9am. Though our team could have chosen to use closing data on an earlier date which could have been more stable, the team chose to use the earlier-announced plan of using the freshest stock market data for the sake of consistency.

On November 9, Adida and Zagórski independently confirmed that Scantegrity correctly responded to all digital challenges. In particular, they confirmed that the tally computation audit data was correct. Both made available independently-written code on their websites that voters and others could use to check the tally computation commitments. The chair mentioned that several voters had shown an interest in running the code made available by

Adida and Zagórski, and that she expected that Takoma Park voters used the code to perform some audits themselves.

Confirmation Codes and Complaints

The period for complaints regarding the election (including complaints about missing confirmation numbers) expired at 6 pm on November 6. The Scantegrity website recorded 81 unique ballot ID verifications, of which about 66 (almost 4% of the total votes) were performed before the deadline. Our team was also told by a board member that at least a few voters checked codes on auditor websites. Both Adida and Zagórski made the confirmation numbers available on their websites after the election.

The number of voters who checked their ballots online before the Takoma Park complaint deadline (66), while not large, was sufficient to have detected (with high probability) any errors or fraud large enough to have changed the election outcome (See Equation 3.9, an adversary who changed 10 ballots would be detected with greater than 99% probability).

Scantegrity received a single complaint by a voter who had trouble deciphering a digit in the code and noted it as "0," while the Scantegrity website presented it as "8." The voter requested that codes be printed more clearly in the future. He also stated that if he were not a trusting individual, he would believe that he had proof that his vote was altered.

All codes for all voted ballots were revealed after the dispute resolution period, and all commitments verified by two independent auditors, Adida and Zagórski. Hence, the probability that the code was in error is very small, albeit non-zero. Our team does not believe the code was in error, and there were no other complaints regarding confirmation numbers.

Print Audits

Zagórski provided an interface allowing Coney to check the commitments opened by Scantegrity in Meeting 4 against the candidate/confirmation-code correspondence on the ballots she audited. In her report [47], she confirmed that the correspondence between confirmation numbers and candidates on all the printed ballots she audited were correctly provided by the interface.

Followup

The board of elections and a team representative met to discuss the election and opportunities for improvement several weeks after the election. Both sides were largely satisfied with the election. Conversations have begun regarding the use of Scantegrity in the next municipal election at Takoma Park in November 2011.

5.4 Observations of Voter Experiences

To understand the flow of voters and poll workers, we timed some of the voters as they voted and informally solicited comments from voters as they left the precinct building. We also collected surveys, analyzed in Chapter 6. Approved by the board of elections and UMBC's Institutional Review Board, our procedures respected the constraint of not interfering with the election process. This section summarizes the results of our observations and surveys.

5.4.1 Timing Data

Sitting unobtrusively as official observers in a designated area of the polling room for part of the day, two helpers (not members of the Scantegrity team) timed 93 voters as they carried out the voting process. Using stopwatches, they measured the number of seconds that transpired from the time the voter received a ballot to the time the voter began walking away from the scanner.

Voting times ranged from 55 seconds to 10minutes (the second longest time was 385 seconds), with a mean of 167 seconds and a median of 150 seconds. On average, voters who appeared older took longer than voters who appeared younger. Most of the time was spent marking the ballot. The average time to vote was significantly faster than during the April 2009 mock election, when voters took approximately 8 minutes on average due primarily to scanning delays [136].

The observers noted that many voters did not fully use the privacy sleeve as intended, removing the ballot before scanning rather than inserting the privacy sleeve with the ballot into the scanning slot. Two of the 93 observed voters initially inserted the privacy sleeve upside-down, causing the ballot not to be fed into the scanner (though the scanner could read the ballot in any orientation). A few ran into difficulties trying to insert the sleeve with one hand while holding something else in the other hand.

5.4.2 Election Day Comments From Voters

As voters left the precinct building, members of the Scantegrity team conducting the written surveys, and a helper (a usability expert who is not a member of the Scantegrity team) solicited comments from voters with questions such as "What did you think of the new voting system?" The helper solicited comments 1:30-3:00pm and 7-8pm. A common response was, "It was easy."

Quite a few voters did not understand that they could verify their votes online and that, to do so, they had to write down the confirmation numbers revealed by their ballot choices. Some explained that they intentionally did not read any instructions because they "knew how to vote." Others failed to notice or understand instructions on posters along the waiting line, in the voting booth, on the ballot, and in the Takoma Park newsletter.

In response, later in the day, we announced to voters as they entered the building that there was a new system and that to verify their votes, they would need to write down their confirmation numbers. These verbal announcements seemed to have some positive effect, and there were fewer voter comments expressing lack of awareness of the verification option after we began the announcements. Nevertheless, some voters were still unaware of the verification option. It was a humbling experience to see firsthand how difficult it can be to get across the most basic points effectively, especially the first time a new system is used.

Some of the voters complained about the double-ended pen: they did not know which end to use or had trouble writing in candidates with the chisel-point (the narrow point was intended for write-ins). A small number of voters had difficulty seeing the confirmation numbers, perhaps because repeatedly pressing too hard could erode the paper. A few voters expressed concern about the difficulty of writing down the confirmation numbers if the ballot had been much longer or had a large number of competing candidates.

Many voters expressed a strong confidence in the integrity of elections, while a small minority expressed sharp distrust in previous electronic election technology. These feelings seemed to be based more on a general subjective belief rather than detailed knowledge of election procedures and technology. Similarly, those expressing strong confidence in Scantegrity seemed to like the concept of verification but did not understand in detail why Scantegrity provides high outcome assurance.

5.5 Discussion and Lessons Learned

Overall, this project should be deemed a success: the goals of the election were met, and there were no major issues. Many aspects of the Scantegrity design and implementation worked well; however, some could be improved in future elections.

5.5.1 Technology Challenges

Perhaps the most challenging aspect for future elections is the scaling of ballot printing. The printers we used were not very reliable.

Unfortunately, we also introduced an error in the generation when switching from alphanumeric to numeric confirmation numbers as a result of findings in the Mock election (see Chapter 4). This resulted in approximately 8.5 bits of entropy as opposed to the expected 10 bits. We discovered this error after we started printing and it was too late to regenerate the audit trail.

The error increased the chance that an adversary could guess an unseen confirmation number to approximately one in 360 rather than the intended one in 1000; a small decrease in the protection afforded against malicious voters trying to guess unseen codes in order to discredit the system.

The website, while sufficient, might utilize existing research in distributed systems to reduce the expectations of observers and voters. The scanner could also be improved with more sophisticated image analysis and increased ability to handle unreadable ballots. It only occurred to us after the election that the write-in resolution process could have greater utility if it were expanded to deal with unreadable and unclear ballots.

Variations on the design that are worth exploring include printing voter receipts (rather than having voters copy confirmation numbers by hand)—there are clearly security aspects to handle if one does this, which are discussed in Chapter 7. The design should also be extended for better accessibility. The special pen might be improved by having only a single medium-tip point rather than two tips of different sizes. The scanning operation and its interaction with the privacy sleeve should be studied and improved.

5.5.2 Real-World Deployment of Research Systems

As is common with many projects, too much was left until the last minute. Better project management would have been helpful, and key aspects should have been finalized earlier. Materials and procedures should be more extensively tested beforehand.

One of the most important lessons learned is the value of close collaboration and clear communication between election officials and the election system providers, whether they are researchers or vendors.

Another lesson learned is that it is important to provide voters with clear explanations of the new features of a voting system and to do so efficiently, with minimal impact on throughput. Resolving the tension between these requirements definitely needs further exploration. For example, it might be worthwhile to have an instructional video explaining the Scantegrity system that voters could watch as they come in. The permanent adoption of Scantegrity II in a jurisdiction would, however, alleviate the educational burden over time, as voters learn the system's features in successive elections.

5.5.3 Comparison with Post-Election Audits

It is interesting to compare Scantegrity with the other major technique for election outcome verification: post-election audits. Because these audits do not allow anyone to check that a particular ballot was counted correctly, they do not provide the level of integrity provided by Scantegrity.

Post-election audits, even those with redundant digital and physical records such as optical scan systems, only address errors or malfeasance in the counting of votes and not in the chain of custody.[15] In contrast, E2E voting systems such as Scantegrity provide a verifiable chain of custody. Voters can check that their ballots are included in the tally, and anyone—not just a privileged group of auditors—can check that those ballots are tallied as intended.

It must be admitted, however, that the additional integrity benefits provided by Scantegrity II come at the cost of somewhat increased complexity and at the cost of an increased (but manageable) risk to voter privacy (since ballots are uniquely identifiable). That said, some jurisdictions and/or election systems require or use serial numbers on ballots, and we have proposed several possible approaches to destroy or obfuscate serial number information appropriately. Furthermore, it can be argued that a voter wishing to "fingerprint" a ballot can do so without being detected in current paper ballot systems simply by marking ovals in distinctive ways.

[15] Having multiple records may make an attacker's job harder, but note that the attacker only has to change the record that will ultimately be used and/or trusted (not necessarily both). Also, redundancy can work against a system, as changing a digital record in an obviously malicious way may allow a more subtle manipulation of the physical record.

Chapter 6

Analysis of Election Survey Data

On election day in November 2009, we surveyed voters and election judges about their experiences using and administering the new voting system. In this chapter we report our findings, providing an assessment of how voters and election officials react to an E2E system in a binding governmental election. Because we study voters and poll workers in a real election—as opposed to in a simulated election—these findings more accurately reflect the true experiences of voters and poll workers.

Our study provides insight into the experience of a binding election, into the use of an end-to-end-verifiable system in general, and Scantegrity in particular. We hypothesized that most voters would react as if the system were a traditional optical scan system and be satisfied with the system as seen in other studies, and that the verification mechanism would provide meaningful feedback to increase confidence in the system. We examine the following questions:

1. Is the ability to verify votes valued by voters?

2. Is the additional layer of verifiability suitably transparent to voters and poll workers?

3. Does receipt creation impact the voter experience?

4. Do voters accept the benefits of a system even if they do not necessarily completely understand the underlying technology that drives it?

5. Does the system impact voters of any particular demographic (such as with regard to computer expertise, income or education)?

This election study is the first to study an E2E system with ballot privacy as used in a binding governmental election. Additionally, among studies of E2E systems it is the first to survey such a diverse group of voters in a binding election, and the first to survey election judges in a binding election. With regard to other election studies, it is among only a handful that study voters in a binding election. Binding elections necessarily constrain research methodologies because only one voting system may be used and errors must not be intentionally introduced. Therefore some statements cannot be addressed as well as we would like to and some other statements are out of scope (e.g., a comparison study). Nevertheless, a study of a binding election offers significant advantages over that of the mock election and other studies, by surveying real voters under real-world conditions which can yield observations more representative of the true voter or poll worker experience.

Although the reaction to Scantegrity was positive, especially by voters, the survey also captures the consequence of a number of procedural missteps and shortcomings of the system implementation. We did not find evidence that aspects of the cryptographic protocol negatively affected voter perception. We found little evidence that demographic factors affected voter experience.

Section 6.1 discusses how we collected our data. We present results in Section 6.2, and a discussion in Section 6.3.

6.1 Methodology

We now describe the research procedures used to collect data during the election. Our research protocols and questionnaires were approved by UMBC's Institutional Review Board, as required for experiments with human subjects. Voters were polled from 8am to 5pm, and voting hours were from 8am to 8pm. The study participants comprised election judges who administered the election and voters who voted in the election.

6.1.1 Research Protocol

Our research team was not permitted in or around the voting area. Instead, we were allowed to use the area typically designated for exit polling. A surveyor was posted at the main exit, and additional surveyors also covered other exits throughout part of the day.

As each voter left the polling location, a surveyor asked the voter if she would be willing to fill out a questionnaire. If yes, the researcher handed the voter a conventional clipboard with two one-sided questionnaires: a voter field test questionnaire and a demographics questionnaire. Form numbers linked the field test and demographics questionnaires filled out by the same voter.

Voters could visit the on-line verification web site after polls closed. We wrote an on-line questionnaire, but Takoma Park requested we put it on the bottom of the ballot check results page. No voters filled out the on-line questionnaire.

Election judges were also given an election judge field test questionnaire and demographics form. At the end of the day, each judge was provided with an addressed, stamped envelope and the survey form. We requested that each judge fill out and mail back the form.

Election judges operated the system entirely by themselves, and select members of our research team were designated to fill vendor roles. These team members worked separately from the surveyors and did not perform any surveyor functions.

The demographics forms and questionnaires were sequentially numbered when given to participants. The sequential numbering allowed us to correlate demographic information with survey responses through the serial number while maintaining the anonymity of the participants. After collecting the forms we scanned the demographic and field study questionnaires separately. Using the sequential ID numbers we entered each response into one row of a spreadsheet using the mark position number to denote the selections of each respondent. Afterward, we imported the spreadsheet into an SQLite database,[1] and used the R statistical language to process the form data.[2]

We did not collect any personal identifying information for voters or election judges. We did however, serially number the forms handed out and the demographics forms were keyed to demographics forms. In some cases, where judges or voters fall into specific demographic patterns, or when the time the voters filled out the survey is known, it may be possible to identify some respondents.

6.1.2 Research Instruments

We gave out three types of forms to participants. Voters received a demographics form and a voter field study questionnaire. Election judges received the same demographics form, and an election judge field study questionnaire.

[1] http://sqllite.org/

[2] http://r-project.org

Demographics Form

We gave voters and election officials the same demographics form, consisting of 12 questions on standard demographic information and information about prior experience with different types of voting systems. The questions asked respondents about their sex, age, race, languages spoken, education level, computer usage, participation in previous elections, whether any mistakes had been made while voting in previous elections, previous voting systems used, physical challenges, and annual household income. See Figure 6.1 to view the demographics form.

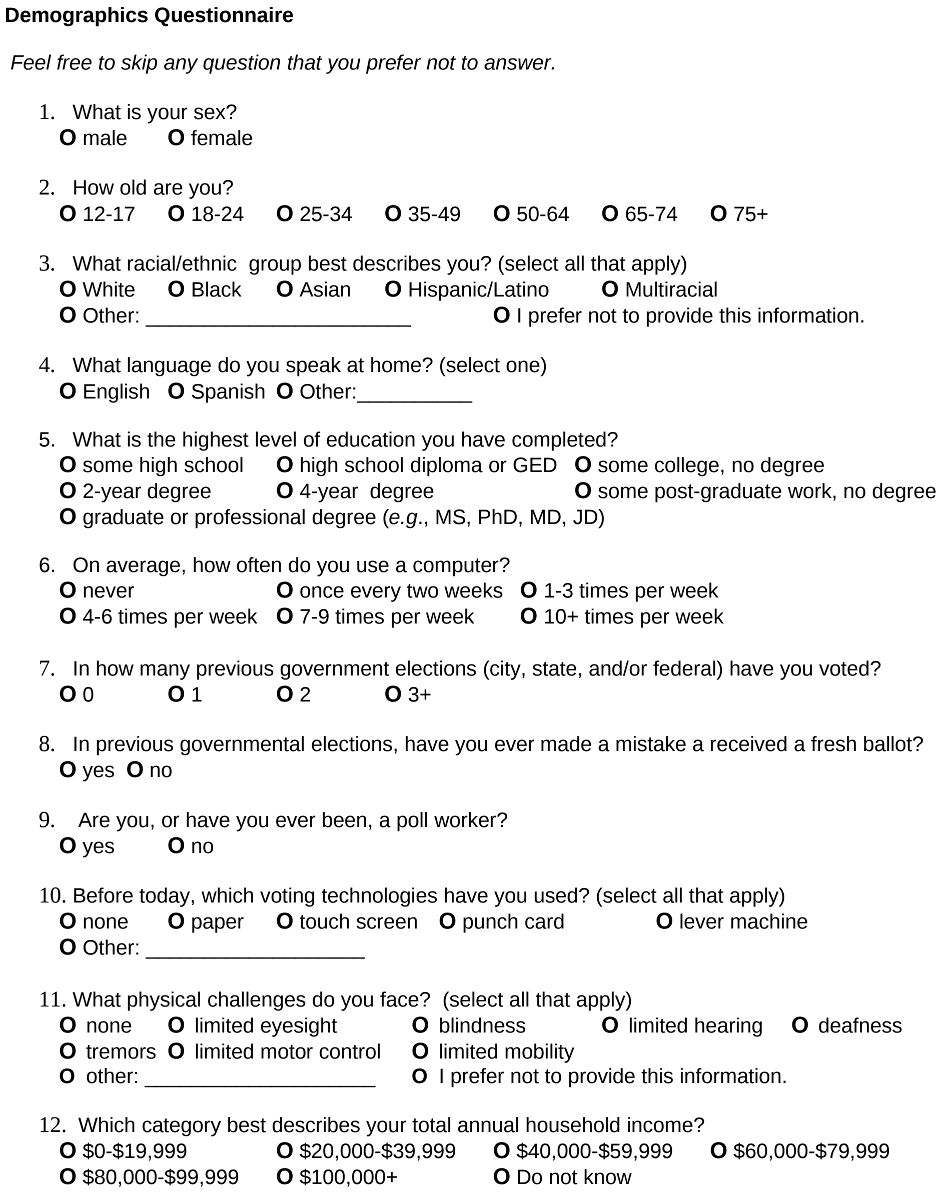
Voter Field Study Questionnaire

The voter questionnaire comprised 17 questions. The first 12 used a 7-point Likert (Strongly Disagree to Strongly Agree) scale, and included a "not applicable" option. [3] The remaining questions (14 through 18) asked respondents how many times they made mistakes, whether they attempted to audit ballot printing, whether they asked for assistance, whether they had any difficulties voting, and whether they had comments about the process. Figure 6.2 shows the questionnaire used during the election.

Election Judge Field Study Questionnaire

The election judge questionnaire was two-sided. The first side featured fourteen 7-point Likert scale questions, which asked whether the system was easy to administer, whether it was easy for voters to mark ballots, whether it was easy for voters to correct mistakes, whether it was easy for voters to record code numbers, whether the voting system was easy for voters to use, whether voters appeared comfortable using the system, whether the

[3] A typo on the survey skipped the numbering on the voter questionnaire from 2 to 4. This error is preserved in this paper.



Demographics Questionnaire

Feel free to skip any question that you prefer not to answer.

1. What is your sex?
○ male ○ female

2. How old are you?
○ 12-17 ○ 18-24 ○ 25-34 ○ 35-49 ○ 50-64 ○ 65-74 ○ 75+

3. What racial/ethnic group best describes you? (select all that apply)
○ White ○ Black ○ Asian ○ Hispanic/Latino ○ Multiracial
○ Other: ______________________ ○ I prefer not to provide this information.

4. What language do you speak at home? (select one)
○ English ○ Spanish ○ Other:__________

5. What is the highest level of education you have completed?
○ some high school ○ high school diploma or GED ○ some college, no degree
○ 2-year degree ○ 4-year degree ○ some post-graduate work, no degree
○ graduate or professional degree (*e.g.*, MS, PhD, MD, JD)

6. On average, how often do you use a computer?
○ never ○ once every two weeks ○ 1-3 times per week
○ 4-6 times per week ○ 7-9 times per week ○ 10+ times per week

7. In how many previous government elections (city, state, and/or federal) have you voted?
○ 0 ○ 1 ○ 2 ○ 3+

8. In previous governmental elections, have you ever made a mistake a received a fresh ballot?
○ yes ○ no

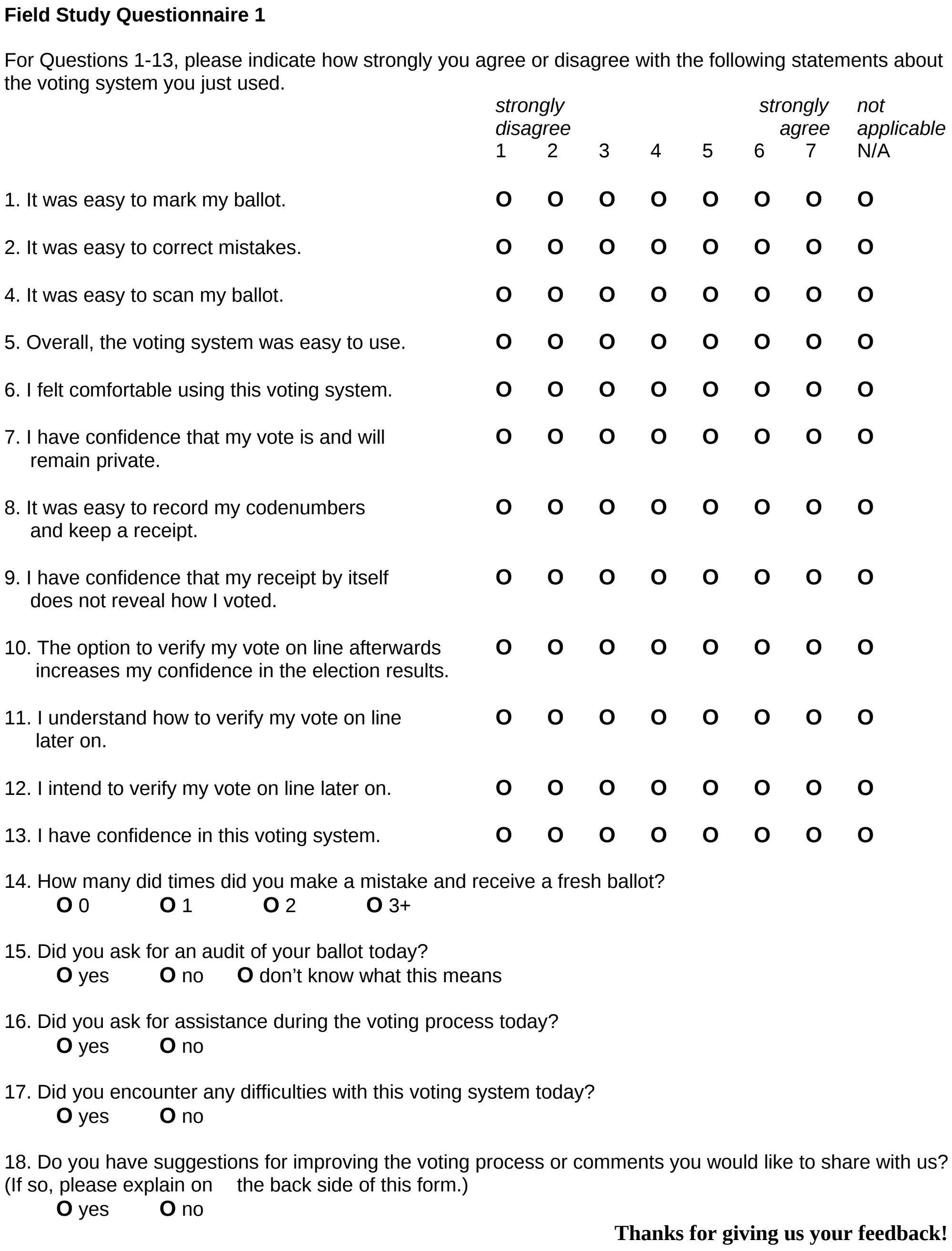
9. Are you, or have you ever been, a poll worker?
○ yes ○ no

10. Before today, which voting technologies have you used? (select all that apply)
○ none ○ paper ○ touch screen ○ punch card ○ lever machine
○ Other: ___________________

11. What physical challenges do you face? (select all that apply)
○ none ○ limited eyesight ○ blindness ○ limited hearing ○ deafness
○ tremors ○ limited motor control ○ limited mobility
○ other: ____________________ ○ I prefer not to provide this information.

12. Which category best describes your total annual household income?
○ $0-$19,999 ○ $20,000-$39,999 ○ $40,000-$59,999 ○ $60,000-$79,999
○ $80,000-$99,999 ○ $100,000+ ○ Do not know

Figure 6.1: The demographics form used during the municipal election.



Field Study Questionnaire 1

For Questions 1-13, please indicate how strongly you agree or disagree with the following statements about the voting system you just used.

	strongly disagree						*strongly agree*	*not applicable*
	1	2	3	4	5	6	7	N/A
1. It was easy to mark my ballot.	○	○	○	○	○	○	○	○
2. It was easy to correct mistakes.	○	○	○	○	○	○	○	○
4. It was easy to scan my ballot.	○	○	○	○	○	○	○	○
5. Overall, the voting system was easy to use.	○	○	○	○	○	○	○	○
6. I felt comfortable using this voting system.	○	○	○	○	○	○	○	○
7. I have confidence that my vote is and will remain private.	○	○	○	○	○	○	○	○
8. It was easy to record my codenumbers and keep a receipt.	○	○	○	○	○	○	○	○
9. I have confidence that my receipt by itself does not reveal how I voted.	○	○	○	○	○	○	○	○
10. The option to verify my vote on line afterwards increases my confidence in the election results.	○	○	○	○	○	○	○	○
11. I understand how to verify my vote on line later on.	○	○	○	○	○	○	○	○
12. I intend to verify my vote on line later on.	○	○	○	○	○	○	○	○
13. I have confidence in this voting system.	○	○	○	○	○	○	○	○

14. How many did times did you make a mistake and receive a fresh ballot?
○ 0 ○ 1 ○ 2 ○ 3+

15. Did you ask for an audit of your ballot today?
○ yes ○ no ○ don’t know what this means

16. Did you ask for assistance during the voting process today?
○ yes ○ no

17. Did you encounter any difficulties with this voting system today?
○ yes ○ no

18. Do you have suggestions for improving the voting process or comments you would like to share with us? (If so, please explain on the back side of this form.)
○ yes ○ no

Thanks for giving us your feedback!

Figure 6.2: The voter questionnaire used during the municipal election.

respondent had confidence ballots were correctly printed, whether the official data recorded votes as intended, whether the final tally correctly included votes as case, whether the receipt revealed how the voter voted, whether votes were private and remained so, whether the option to verify the vote increases their confidence, and whether they had confidence in the voting system. The second side of the sheet asked free-form questions about difficulties administering the voting system, difficulties observed that voters had using the voting system, suggestions to improve the voting process, and additional comments. Figures 6.3 and 6.4 show the judge questionnaires.

6.1.3 Interpretation of Raw Data

Due to the nature of our study, participants were often unclear in their responses and we were unable to seek clarification from them. We had to interpret some of the responses, and we endeavored to do so in a clear and consistent fashion:

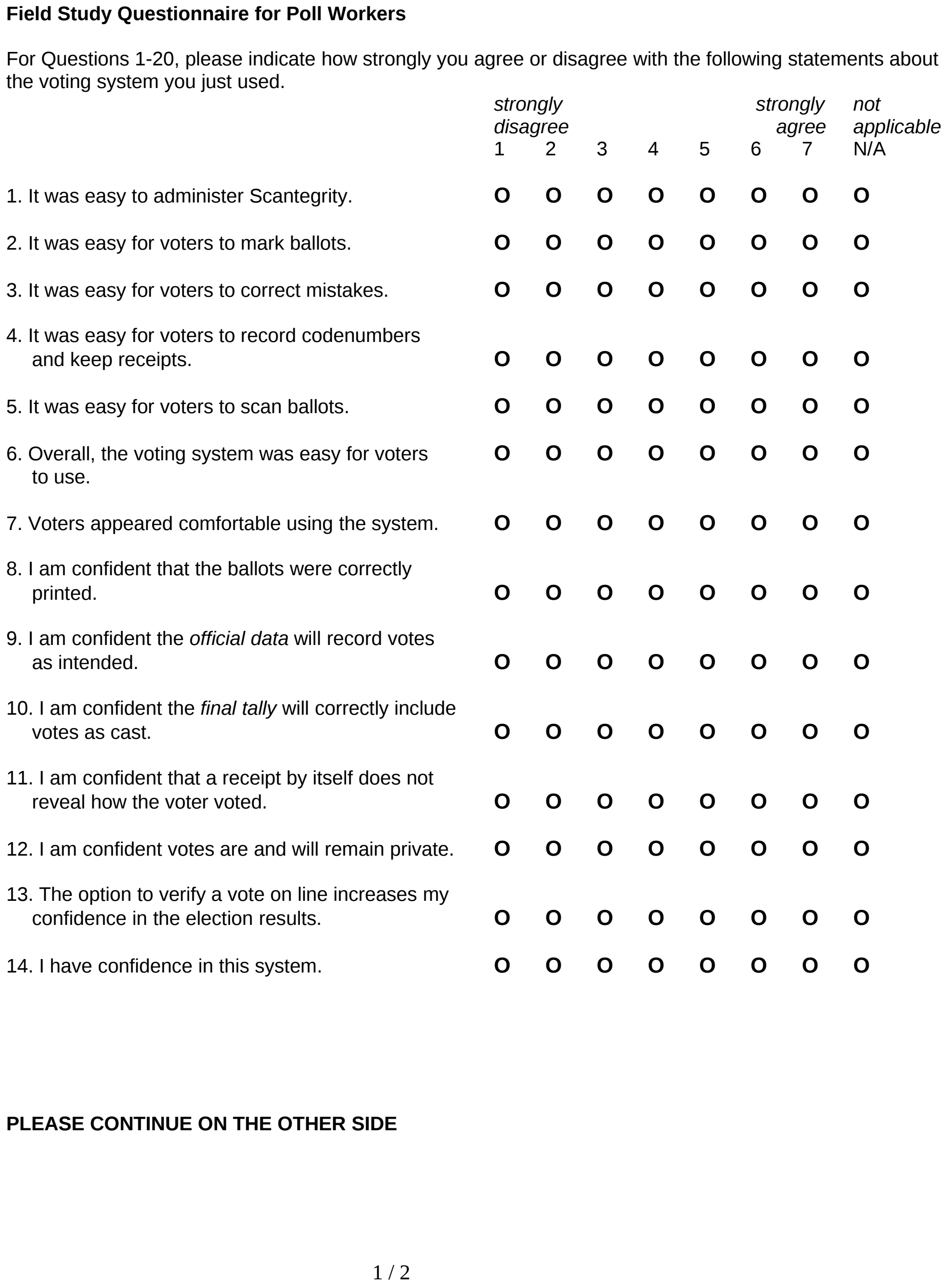
- If a respondent marked multiple answers on questions, either the darkest response or the response most like the other responses was accepted.

- If the respondent crossed out the unintended response to make his or her intent clear, we took that response instead of the darkest.

We recorded in our spreadsheet a column for each response describing any intent issues and how they were resolved.

6.1.4 Actions In the Polling Site

Our team was allowed to observe election day events, but we were not permitted to serve as election judges nor to interfere with the elections process. Additionally, only two representatives were permitted in the voting area at any given time.



Field Study Questionnaire for Poll Workers

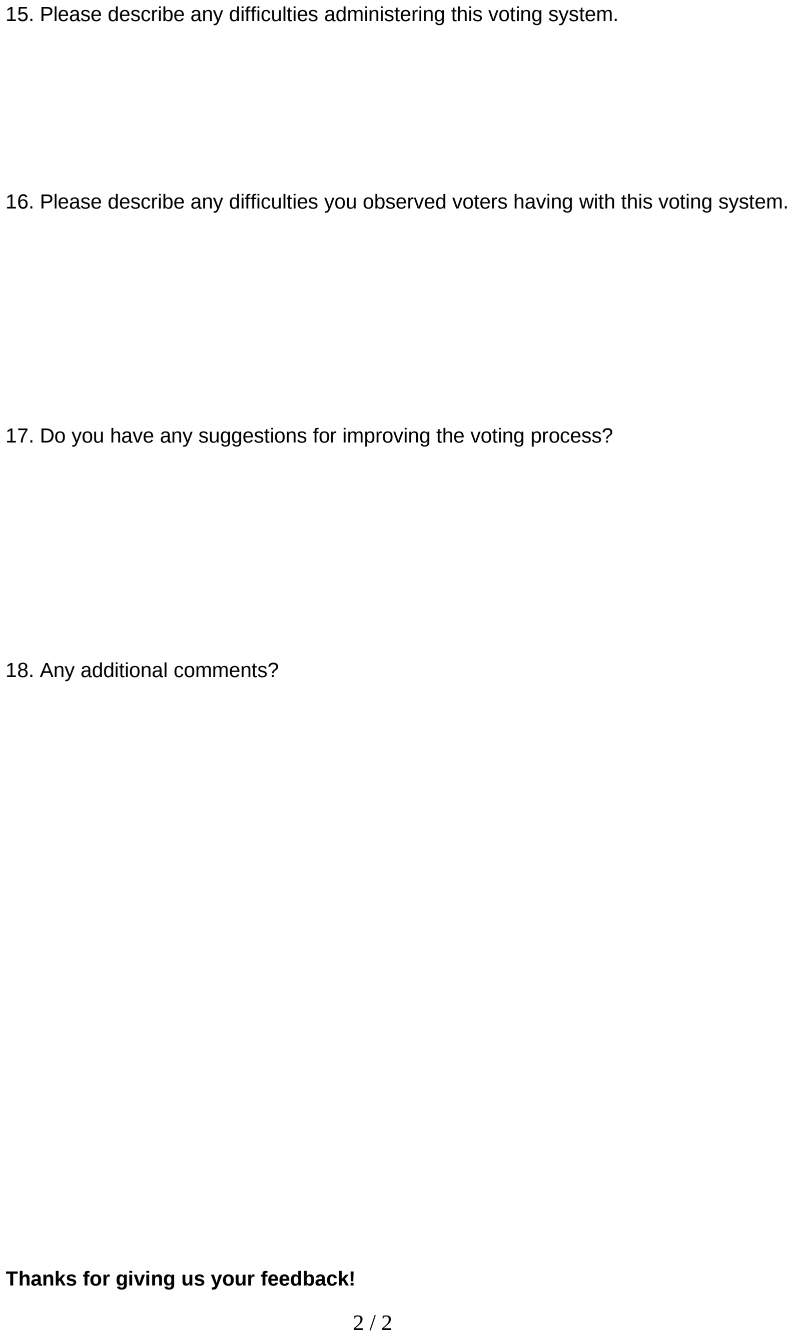
For Questions 1-20, please indicate how strongly you agree or disagree with the following statements about the voting system you just used.

	strongly						*strongly*	*not*
	disagree						*agree*	*applicable*
	1	2	3	4	5	6	7	N/A
1. It was easy to administer Scantegrity.	○	○	○	○	○	○	○	○
2. It was easy for voters to mark ballots.	○	○	○	○	○	○	○	○
3. It was easy for voters to correct mistakes.	○	○	○	○	○	○	○	○
4. It was easy for voters to record codenumbers and keep receipts.	○	○	○	○	○	○	○	○
5. It was easy for voters to scan ballots.	○	○	○	○	○	○	○	○
6. Overall, the voting system was easy for voters to use.	○	○	○	○	○	○	○	○
7. Voters appeared comfortable using the system.	○	○	○	○	○	○	○	○
8. I am confident that the ballots were correctly printed.	○	○	○	○	○	○	○	○
9. I am confident the *official data* will record votes as intended.	○	○	○	○	○	○	○	○
10. I am confident the *final tally* will correctly include votes as cast.	○	○	○	○	○	○	○	○
11. I am confident that a receipt by itself does not reveal how the voter voted.	○	○	○	○	○	○	○	○
12. I am confident votes are and will remain private.	○	○	○	○	○	○	○	○
13. The option to verify a vote on line increases my confidence in the election results.	○	○	○	○	○	○	○	○
14. I have confidence in this system.	○	○	○	○	○	○	○	○

PLEASE CONTINUE ON THE OTHER SIDE

1 / 2

Figure 6.3: Page 1 of the election judge questionnaire used during the municipal election.



15. Please describe any difficulties administering this voting system.

16. Please describe any difficulties you observed voters having with this voting system.

17. Do you have any suggestions for improving the voting process?

18. Any additional comments?

Thanks for giving us your feedback!

2 / 2

Figure 6.4: Page 2 of the election judge questionnaire used during the municipal election.

Two members of our team acted as technical support when needed, fulfilling the role that a vendor would during election day.[4] Before the election the technical support team was directed to set up the scanning stations under supervision by an election judge. After the election they were asked to disconnect the scanning stations and to collate the memory sticks for tabulation by the election night tabulation software. The technical support team members did not interact with participants in the survey, and did not instruct the election judges regarding the survey.

6.2 Results

Descriptive statistics provide evidence which supports our hypothesis that the voting system was positively received by voters. Election judges, however, noted the effort made by voters on several different aspects of the system, and made suggestions for improvements. Voter comments indicated that voters would like more instruction on how to use the voting system.

6.2.1 Voter Surveys

1723 people voted on election day, of whom 276 (16%) filled out surveys. 36 (13%) left the demographics form blank and filled out the questionnaire only. Conversely, 5 (1.8%) respondents left the questionnaire blank but filled out the demographics form. Many did not answer a subset of questions on one or both forms. There were 235 (85.1%) respondents who answered some questions on both sheets, 240 (86.9%) who answered the demographics, and 271 (98.2%) who answered the questionnaire.

[4] In Maryland, technical support representatives from the election vendor are available to election judges at each polling site on election day.

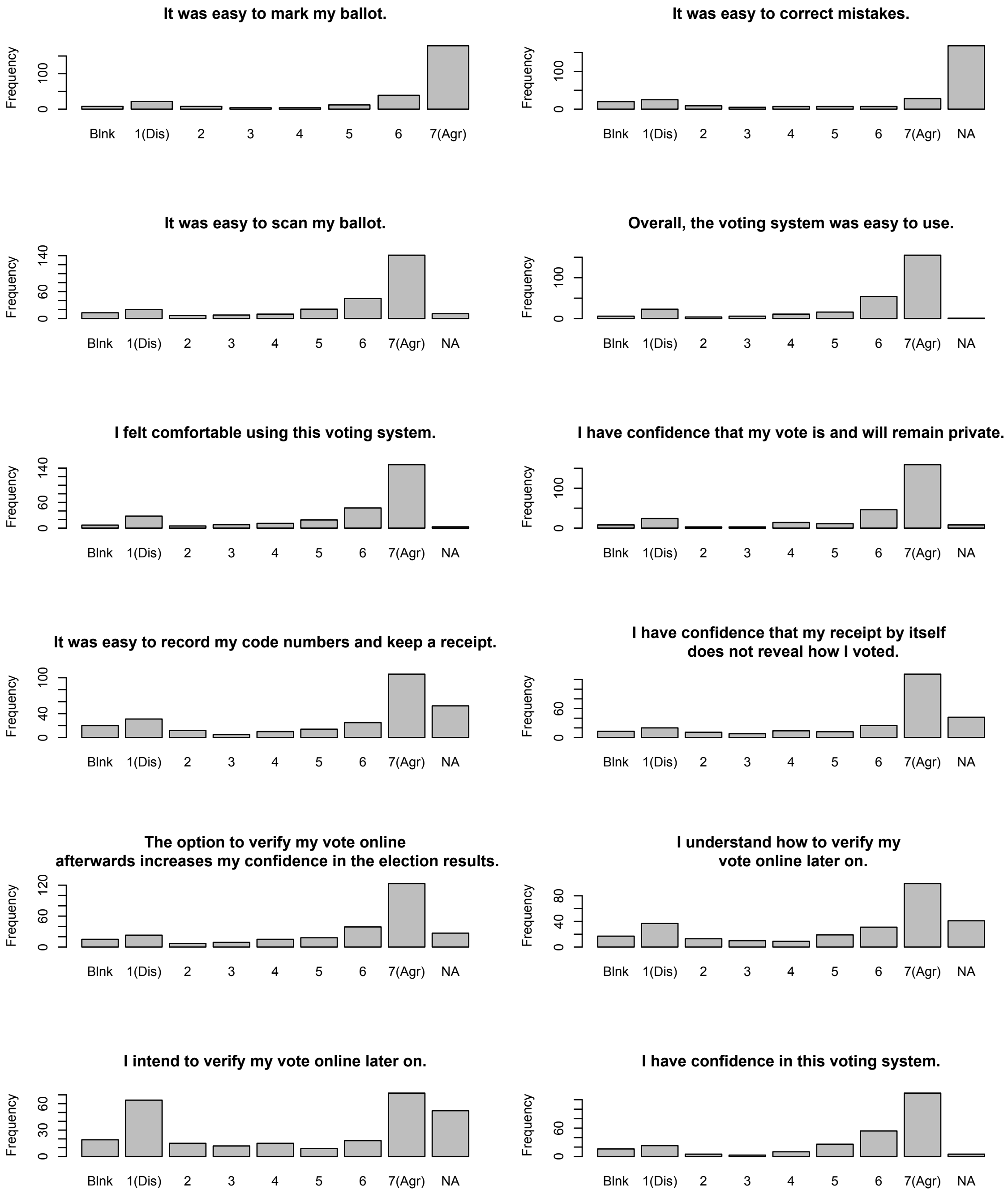
We provide bar charts for voter responses to the questionnaire in Figures 6.5 and 6.6. All of the Likert scales appear bimodal, indicating that respondents were highly opinionated. This is not something we saw in the mock election test study about this system with volunteer users [136]. Figure 6.7 shows an alternate view of the Likert questions designed to show correlation. Here we see that general agreement/disagreement of respondents on each question varies. Q2 (EasyToCorrectMistakes) and Q12 (IntendToVerify) appear to be equally distributed between the high and low values of the Likert scale.

Voter responses support our general hypothesis of satisfaction for the new voting system. The response to Q13 (HaveConfidence) showed 230 out of 268 participants (85.8%) marked at or above 5 (Somewhat Agree) on the scale, with a mean of 5.78 and a median of 7. Table 6.1 summarizes this information for the rest of the likert questions.

	≥ 5 (Agree)	≤ 3 (Disagree)	Mean	Median	Std. Dev.	N
Q1	230 (85.82%)	34 (12.69%)	6.02	7	1.86	268
Q2	42 (47.73%)	39 (44.32%)	4.08	4	2.52	88
Q4	207 (82.14%)	35 (13.89%)	5.79	7	1.87	252
Q5	225 (83.64%)	33 (12.27%)	5.88	7	1.84	269
Q6	214 (80.45%)	41 (15.41%)	5.71	7	1.98	266
Q7	216 (83.08%)	30 (11.54%)	5.92	7	1.86	260
Q8	145 (71.43%)	48 (23.65%)	5.28	7	2.30	203
Q9	168 (76.02%)	39 (17.65%)	5.65	7	2.05	221
Q10	180 (76.92%)	39 (16.67%)	5.59	7	2.00	234
Q11	149 (68.35%)	60 (27.52%)	5.06	6	2.34	218
Q12	99 (48.29%)	91 (44.39%)	4.13	4	2.59	205
Q13	219 (84.23%)	31 (11.92%)	5.82	7	1.86	260

Table 6.1: Voter responses to Likert scale questions about Scantegrity at the municipal election.

When compared with Q10 (VerificationGivesMoreConfidence), Q12 (IntendToVerify) indicates that respondents said they were more confident in the system because of the receipts, but fewer were willing to check their ballot online at home. This suggest that while



It was easy to mark my ballot.

It was easy to correct mistakes.

It was easy to scan my ballot.

Overall, the voting system was easy to use.

I felt comfortable using this voting system.

I have confidence that my vote is and will remain private.

It was easy to record my code numbers and keep a receipt.

I have confidence that my receipt by itself does not reveal how I voted.

The option to verify my vote online afterwards increases my confidence in the election results.

I understand how to verify my vote online later on.

I intend to verify my vote online later on.

I have confidence in this voting system.

Figure 6.5: Voter reactions to Scantegrity at Takoma Park.

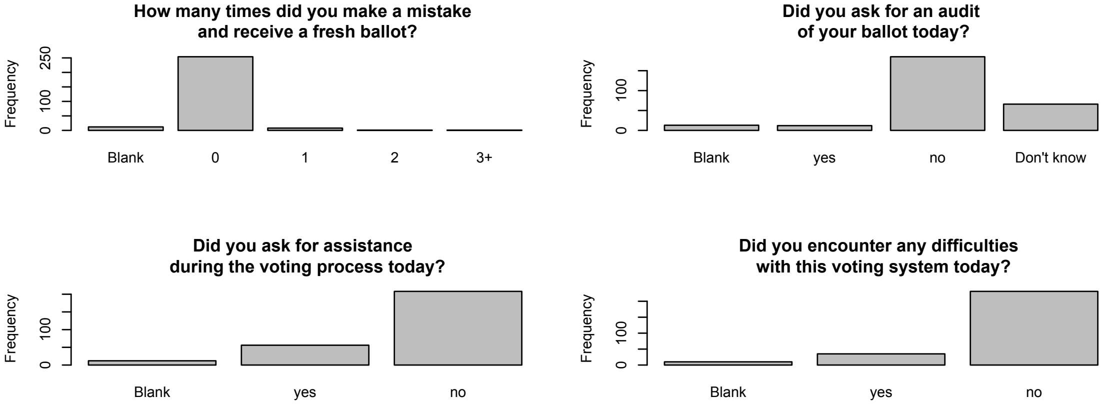


Figure 6.6: Voter reactions to Scantegrity at Takoma Park (cont).

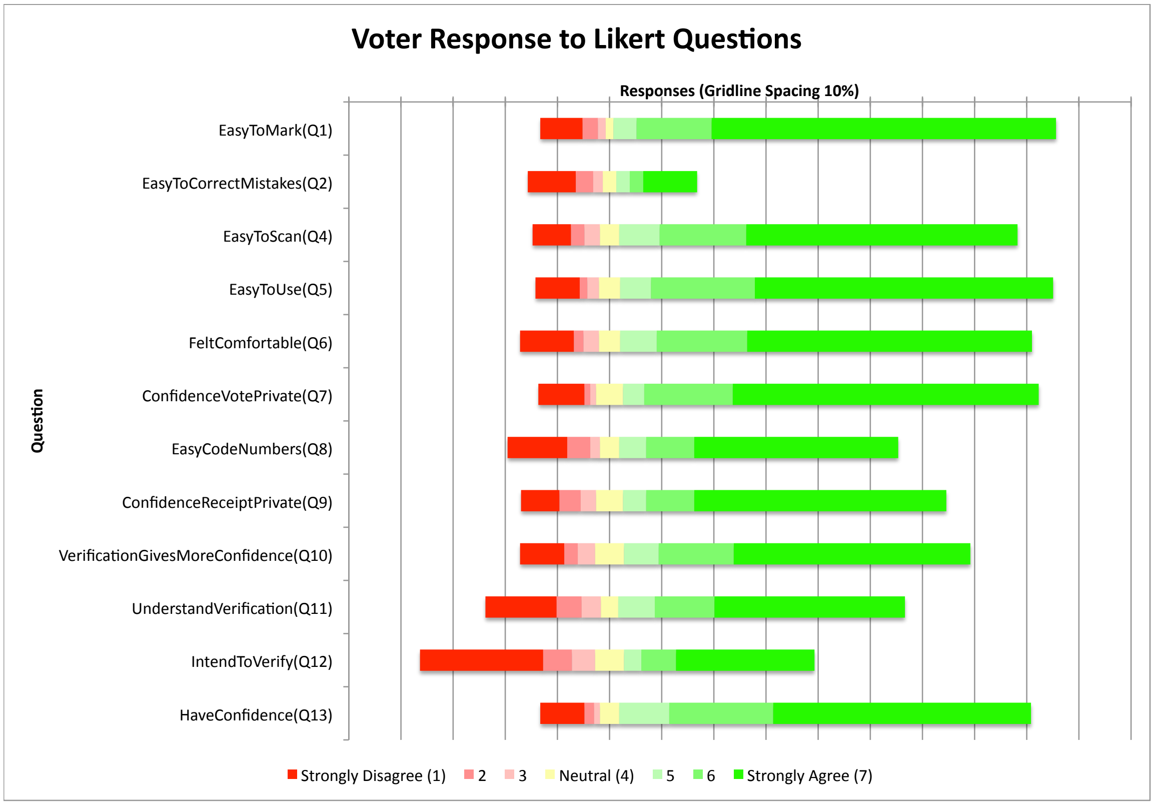


Figure 6.7: Horizontal bar charts showing distributions around the neutral position of the Likert questions. The width of each bar represents the total number of respondents for that question, and each bar is divided into subsections whose width represent the respondents in that category.

voters may find value in the receipts, they might not take advantage of it, which is further supported by our observation that only 81 receipt checks were made by voters after the election (see Section 5.3.3). It also provides evidence that most voters will accept systems that they do not necessarily completely understand.

To understand if voter demographics affect voter experience we use ordinary least squares (OLS) regression of a combined variable we call satisfaction over the demographics factors we collected. To create the satisfaction variable, we combined the average responses to Q1, Q4-11, and Q13 for each respondent. We dropped Q2 because it appears that most voters misread the question to put NA because they did not make a mistake (168 out of 256 respondents put NA), and we dropped Q11 because it deals with expected behavior and not satisfaction with the process. The cronbach's α of the remaining selected questions was .97 (N = 142).

The resulting dependent satisfaction variable had a mean of 5.69[5] (StdDev = 1.7, N = 271). Because the data was negatively skewed (-1.94) and had high kurtosis (5.92) we analyzed the cube (x^3) of the values (skew = -.86, kurtosis = 2.69).

Age, education, computer use, and income were coded as ordinal data starting at 0 for the smallest category (see Figure 6.1 for the specific categories and range of each variable). Gender was coded to 1 for female, 0 for male. We coded 1 if the respondent was black non-hispanic, a former election judge, reported any disabilities (except hearing), or used any of a touch screen, punch card, or lever machine system.

We expected positive effects for education, computer use, being a former election judge, black non-hispanics, and use of any type of voting machine. People who have more education and use the computer more often might be more likely to understand the system,

[5] A table depicting mean scores of the satisfaction index for all demographic factors is available in figure A.3 in appendix A.

and thus see its value. Black non-hispanics have been observed to react positively to optical scan systems in other studies [75]. People who have been election judges or who have used different voting equipment might be more familiar with deficiencies in existing equipment and might also be more likely to see value in the ability to take home a receipt.

We expected negative effects for age, non-english speakers, and people with disabilities. We believed older voters would dislike the internet-enabled part of the system, and that people with disabilities would dislike the paper. Non-english speakers are at a natural disadvantage. We did not expect to see significant impact by gender or income.

Table 6.2.1 provides results for the effects of demographic variables on satisfaction. The model produces a mix of agreement and disagreement with our hypotheses.

The model shows a statistically significant positive effect for females and negative effect for income, and we did not expect either of these variables to have statistically significant effects. The income effect is smaller and weaker compared to the other significant variables, and could be the result of a sample size that is too small. It is unclear why women would react more favorably to the system.

While the effect of punch cards was statistically significant in the expected direction, the effect of touch screens was in the opposite direction. We suspect this is a result of the population being highly computer literate (almost 73% reported using the computer 10+ times a week) and already accustomed to the existing touch screen DRE system currently in use in Maryland.

The rest of the variables in the model did not show significant effects on satisfaction. The age, black non-hispanic, non-english, education, and disabilities variables show effects in the expected direction. The computer use, former judge, and lever variables did not show effects in the expected direction. The former judge variable, in particular, showed a strong

Coefficients:	Estimate	Std. Error	t value	Pr(> $\|t\|$)	
Intercept(Const)	274.162	62.588	4.380	1.98e-05	***
Gender	33.535	15.760	2.128	0.01733	*
Age	-3.515	9.516	-0.369	0.35612	
Black Non-Hispanic	13.448	25.109	0.536	0.29644	
Non-English	-39.987	33.042	-1.210	0.11387	
Education	1.419	5.808	0.244	0.40366	
Computer Use	-1.254	8.083	-0.155	0.43844	
Former Judge	-19.533	21.545	-0.907	0.18290	
Income	-6.885	5.096	-1.351	0.08915	.
Disabilities	-26.540	28.462	-0.932	0.17615	
Touch screen	-47.516	25.960	-1.830	0.03440	*
Punch card	48.322	18.425	2.623	0.00472	**
Lever	-7.497	18.135	-0.413	0.33989	
N	199				
Res. SE	106.771				
Mult. R^2	0.0945				
Adj. R^2	0.0361				
Signif. codes:	'***' 0.001	'**' 0.01	'*' 0.05	'.' 0.1	

Table 6.2: Voter satisfaction with Scantegrity across selected characteristics. Entries are OLS coefficients, standard errors, t-values, and probabilities. All entries are one-tailed, except Intercept(Const).

negative effect. We believe these voters were more likely to notice less polished aspects of the system.

We further explored the demographics effects using correlation analysis, and we found women tended to agree more strongly than men on Q1(EasyToMark) ($\chi^2 = 10.14$, $p = 0.001$, $df = 1$), Q4(EasyScan) ($\chi^2 = 4.18$, $p = 0.041$, $df = 1$), Q5(EasyUse) ($\chi^2 = 7.64$, $p = 0.006$, $df = 1$), Q6(FeltComfy) ($\chi^2 = 5.0$, $p = 0.025$, $df = 1$), and Q7(ConfPriv) ($\chi^2 = 9.25$, $p = 0.002$, $df = 1$). We did not find correlation of the survey questions with age, race, education, income, computer usage, or experience in previous elections. There were some correlations among demographic data. We include these in Appendix A.1. Correlation tables of the demographic data with the questionnaire are in Appendix A.2.

Voter Comments

51 voters wrote comments on the questionnaires, often pointing out confusion about various aspects of the process:

1. Many were unaware of the verification option.

2. Some did not realize they were supposed to write down confirmation numbers.

3. Some found the pens confusing to use: they did not realize that the pens would expose confirmation numbers, and they did not know which end to use.

4. Some found confirmation numbers were hard to read.

5. Some did not understand how to mark an IRV ballot.

6. Some did not know how to place the ballot into the scanner.

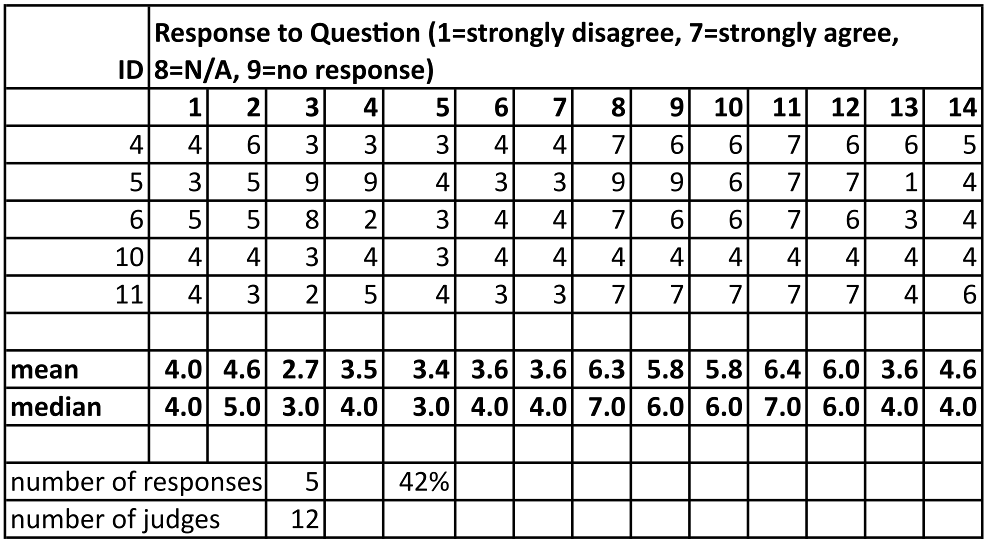
7. One had no difficulty but wondered if seniors or people who speak neither English nor Spanish might have difficulties.

8. One wondered if the government might be able to discern his vote by linking his IP address used during verification with his ballot serial number and noting the time that he was issued a ballot.

9. Many suggested that it would have been helpful to have better instructions, including instruction while they waited in line.

6.2.2 Summary of Election Judge Surveys

There were 12 election judges on election day, of which 5 (42%) responded to our survey. The Judges were much more mixed in their reaction to the new system than were the voters



ID	**Response to Question (1=strongly disagree, 7=strongly agree, 8=N/A, 9=no response)**													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
4	4	6	3	3	3	4	4	7	6	6	7	6	6	5
5	3	5	9	9	4	3	3	9	9	6	7	7	1	4
6	5	5	8	2	3	4	4	7	6	6	7	6	3	4
10	4	4	3	4	3	4	4	4	4	4	4	4	4	4
11	4	3	2	5	4	3	3	7	7	7	7	7	4	6
mean	**4.0**	**4.6**	**2.7**	**3.5**	**3.4**	**3.6**	**3.6**	**6.3**	**5.8**	**5.8**	**6.4**	**6.0**	**3.6**	**4.6**
median	**4.0**	**5.0**	**3.0**	**4.0**	**3.0**	**4.0**	**4.0**	**7.0**	**6.0**	**6.0**	**7.0**	**6.0**	**4.0**	**4.0**
number of responses			5		42%									
number of judges			12											

Figure 6.8: Election judge reactions to Scantegrity at Takoma Park.

in our sample. Figure 6.8summarizes responses to the Likert questions on the election judge survey.[6]

The judges noted the following difficulties:

1. There was too much information.
2. Some voters did not understand what to do, including how to create a receipt.
3. Some voters did not understand how to mark an IRV ballot.
4. The privacy sleeve was hard to use with one hand.
5. The double-ended pens created confusion.
6. Voters, poll workers, and the Scantegrity team have different needs.
7. One wondered whether Scantegrity was worth the extra trouble.

The judges offered the following suggestions:

[6] We chose not to show election judge demographics to protect privacy.

1. Simplify the ballot.

2. Provide receipts so that voters do not have to copy confirmation numbers.

3. Develop better pre-election voter education.

6.3 Discussion

In Chapter 1 we explain our interest in exploring how voters and election officials react to Scantegrity and its use of codes to represent votes. In particular, we are interested in determining if the ability to verify votes is valued by voters, and also whether this additional layer of verifiability is suitably transparent to voters and poll workers. The data from our sample indicate that voters have high levels of appreciation for Scantegrity, that there is no evidence the extra functionality detracts from voter experience, and that voters say the verification function it provides increases confidence in election results. Election officials were less enthusiastic about the system but they did not report that the system overhead too intensive.

We also obtained answers in more specific areas. The system was not too complex to use and administer in the context of elections at Takoma Park. Election judges responding to the survey found it workable. Comments from election judges and voters either spoke well of or ignored the verification option and focused on other issues they perceived to be problematic in the election. Large majorities of voters found various aspects of the system easy to use. 145 out of 203 respondents (71.4%) agreed to the statement "It was easy to record my code numbers and keep a receipt."

The small number of questions on each ballot worked in the system's favor. Increasing the number of questions on the ballot would likely decrease impressions of usability, although this is, to varying degrees, the case for any system.

The extra work involved to participate in verification does not appear to negatively impact voter experience. Levels of confidence for the system were high, indicating that the impact could be comparable to other systems.

Voters appeared to accept the system even if they did not understand it. This study shows high levels of support for the verification receipt, even when voters indicated they did not understand the cryptographic mechanisms behind it. The understand question correlated with confidence less well than other metrics on the survey, which indicates that voters had confidence and found the system easy to use even if they did not necessarily understand parts of the system (see Figure 6.7).

Respondents did say that they appreciated the extra security as the presence of the verification option was reported by voters to increase their confidence in the system (77% agreed to Q10, $N = 234$). This contrasts with the number of people who reported intent to verify (48.3%, $N = 205$), and more so with 81 checks that actually occurred on the ballot data. It appears that, even though voters appreciated the technology, they did not necessarily care to use it.

It is nontrivial to address if certain voting populations will be disadvantaged by this system because participants might not notice or report problems or difficulties they have with the system. We did not find statistically significant evidence that non-computer users, less education, black non-hispanic, or other factors had an undesirable impact on voter satisfaction. We did find that women tended to agree more strongly than men on several ease of use questions, although men still overwhelming agreed strongly. We also found higher income and prior use of a touch screen to negatively impact satisfaction.

Mathematically and technically speaking, we did get enough voters to verify their ballot for this particular election (see Chapter 5). Unfortunately, we cannot make the argument that we will always have enough verifiers to ensure election security. However, we also cannot find any evidence in the data to support the argument that we will not. This statement should be investigated in future work.

6.3.1 Known Limitations & Recommendations

There are a few limitations with the data we collected. The most significant is that due to the way procedures were implemented many voters were not aware of the receipt function. The receipts were put in each poll booth for voters to find. While instructions were available, the receipts were not explained to the voters as they were given a ballot, and many respondents complained to the surveyors and in comments about this issue. Our most important recommendation is to find ways to increase awareness and the number of individuals who will take home a receipt. The best way to accomplish this goal is to find a way to produce and provide the receipts automatically as part of the election process, which we discuss in Chapter 7.

Another issue is that our survey sampled few voters with disabilities. It is important to understand how this new model will affect these users.

Q2 (EasyToCorrectMistakes) did not yield any useful information. Most voters marked NA or left this question blank, which was the correct response if they did not make any mistakes. We would not ask this in future surveys unless the voter had to experience the process of correcting a mistake. Q12 asks about intention and not experience. In the future we will instead use two binary questions, e.g.,“Did you record a receipt?” and “Do you intend to use your receipt to verify your vote online tomorrow?”

Because the study is observational, we were unable to address the question of the effects of the confirmation number receipts as well as we would have liked. Respondents to the questionnaire reported that the presence of a receipt increased their confidence in the results, but how many would have high confidence in the results if they had also used a system without a receipt? A comparative study which looks closely at this issue is a next step for this research.

There were several miscellaneous technical problems throughout the voting day. In the morning one scanner's power was tripped. Later in the day a few voters reported that the scanners had trouble pulling the ballots out of the privacy sleeves. One voter dropped his receipt into the scanner. One scanner jammed but was quickly fixed. A respondent who was the victim of, or witnessed, any of these issues was likely to have a negative response, and it is impossible to control for these types of issues in a real world environment. We recommend that technical measures be taken to minimize these issues, and we agree with the election judges that steps should be taken to simplify the ballot and to develop better pre-election voter education resources.

We believe that, while voters appreciate the ability to verify, it is important that every aspect of the system work well. Much attention was payed to small details, such as the ballot layout, in this implementation of Scantegrity. Instructions for IRV and how to record a receipt were checked carefully for clarity. Takoma Park used an official staff member to create translations of these instructions. The scanner recognized ballots being placed into the system with a beep. These seemingly unimportant issues led to a reasonably smooth election, but more work could have been done to improve the voter and election judge experiences.

Chapter 7

Scantegrity Receipt Printer

When using Scantegrity, voters had trouble recording the confirmation numbers to make their receipts. Often, voters did not notice that they could make a receipt, or they did not follow the instructions to write down the online verification number and other information so that they can check the receipt online after the election. The codes can also be hard to read, and voters can make mistakes when writing the codes down. We designed two different kinds of receipt printing units to solve these issues.

Adding a receipt printer to Scantegrity can solve these problems by automatically generating receipts for voters, substantially improving use experiences with the system. However, a receipt printer potentially introduces a complex trusted component. The Scantegrity team cautioned against such a trusted component, and has not proposed a design for it. A malicious receipt printer could generate improper receipts that may go unnoticed by voters and cause undetectable changes in election outcomes, and violate voter privacy.

In this chapter, we propose two designs for a receipt printer that balance trust and usability, enabling the benefits of a printer while minimizing privacy risk. The first design attempts to be as simple as possible, and the other provides more features at the cost of additional complexity. Both of our designs make use of the TPM to protect the confidentiality and integrity of information passing through the receipt printer, offering good usability without compromising privacy or election integrity.

Rather than trusting the entire receipt printer *platform*–operating system and software of a general purpose computing platform–we place trust in the *Trusted Computing Group (TCG)*'s Trusted Platform Module (TPM), a small, embedded cryptographic processor that safeguards keys from malicious software. Our designs use the TPM to mitigate the problems of malicious software, enabling the benefits of printing without aiding manipulation, integrity, or privacy attacks. Further, the TPM enables our designs to be trusted to maintain secure digital records of the receipts, enabling bulk, third-party verification of *every* receipt, strengthening the security of the overall Scantegrity approach.

Section 7.1 discusses background information on trusted computing. Section 7.2 presents the functionality and security goals that any receipt printer for Scantegrity must implement, and motivates our design. Section 7.3 provides our two designs, while section 7.4 evaluates the security of these designs. Section 7.5 presents additional considerations for receipt printers.

7.1 Trusted Computing Background

Fink *et al.* gives a comprehensive overview of the TPM relative to the field of voting in [62]. The central TPM features that our designs use include:

- *Platform Configuration Register (PCR)*–storage inside the TPM that securely stores cryptographic hashes of booted printer operating and application software
- *Sealing*–a operation that binds the unwrapping and use of a secret to the identity of the TPM and the software state reflected in the PCRs

- *Monotonic Counters*–non-decreasing counters managed securely within the TPM
- *Quote*–a listing of the PCR values, signed by the TPM
- *Cryptographic Keys*–including the *Attestation Identity Key (AIK)*, a signature key that confirms some *known* TPM without identifying *which specific* TPM, and the *Storage Root Key (SRK)*, the parent of other decryption keys
- *Ownership*–the act of establishing the key hierarchy in the TPM, including creation of the AIK, SRK, and associated keys

Trusted boot is a feature related to the TPM. It is the process by which software components in series make measurements of subsequent components during platform boot, comprising a *boot chain*. In the trusted computing context, the TPM's PCRs store measurements of all components in the chain enabling sealing, quotes, and other operations. An initial component in the computing platform is trusted to make the first measurement of the chain, and is either some firmware in the *Basic Input/Output System (BIOS)*, or on more modern processors such as Intel, the AC_INIT software module verified by codes stored within the CPU.

For other features and references on the TPM, the TCG publishes the main TPM specifications in [73]. Pearson *et al.*give an alternative overview of the TPM and the TCG [111], and Challener *et al.*[31] wrote an excellent practical guide to the TPM for software developers. Developers using the TPM are guided to the TrouSerS software stack and test suite for understanding the programming details [81].

Others have suggested using TPMs for voting. Fink, Sherman and Carback designed a TPM protocol for *Direct Recording Electronic (DRE)* voting that signs the ballot and voter selections using a key managed by the TPM that provides proof of the correct DRE software state at the time the vote was cast [62]. Arbaugh outlined an on-line TPM-based protocol for attesting systems through a central server [9]. Rössler *et al.*suggested hardware security modules for postal-voting [129]. Paul and Tanenbaum [110] sketched a voting system architecture incorporating TPMs. Although there is interest in using TPMs for voting, no previous approach uses TPMs to design secure receipt printers.

7.2 Requirements

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

7.2.1 Functional System Requirements

The functional requirements describe the high level features that any Scantegrity printer must provide:

Printed Scantegrity Codes: The receipt printer will produce Scantegrity receipts, providing the user with the confirmation numbers of the selected candidates and the online verification number for the ballot.

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

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 is good for sighted voters, but designs should accommodate disabled voters and speakers of different languages.

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

7.2.2 Security Goals

In addition to basic features, a Scantegrity receipt printer must uphold some security goals or constraints while providing the basic features:

Privacy: The receipt printer should not compromise voter privacy, e.g.,by disclosing Scantegrity codes to unauthorized parties or 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. Generally speaking, the printer shall not enable an attacker to impart credibility to a false receipt.

Event Control: The receipt printer shall not be capable of presenting valid cryptographic proof prior to, or proceeding, the authorized election period. This requires strong controls on signature keys.

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

There are many ways to meet these requirements and constraints, and a TPM-based approach is merely one way to do this. (A different way might involve provisioning cryptographic material via removable smart cards, but we do not consider such a design because it involves more trusted components than a TPM design.)

7.3 Design

We present two design variations for receipt printing for Scantegrity. The first is a standalone image duplicator, and the second is a marked sense translator that requires state and a connection to the PCOS scanner. There are benefits and problems with each, as will be discussed in Section 6.3.

Both variations assume that the voter has completed her ballot prior to requesting a receipt. Neither variation encodes any information onto the receipt that can identify the voter or how she voted.

Both methods record *attestation evidence* onto the receipt, a special cryptographic code generated by the TPM that can prove to the voter that the receipt printer platform booted the correct software. Since the attestation method is common to both approaches, we present its design as a segue from the basic functional design of the image duplicator to the more feature-rich marked sense translator.

Both designs rely on certain features of the TPM. The central TPM features that our designs use include:

- PCR–storage inside the TPM that securely stores cryptographic hashes of booted printer operating and application software

- *Sealing*–a operation that binds the unwrapping and use of a secret to the identity of the TPM and the software state reflected in the PCR

- *Monotonic Counters*–non-decreasing counters managed securely within the TPM

- *Quote*–a listing of the PCR values, signed by the TPM
- *Cryptographic Keys*–including the AIK, a signature key that confirms some *known* TPM without identifying *which specific* TPM, and the SRK, the parent of other decryption keys
- *Ownership*–the act of establishing the key hierarchy in the TPM, including creation of the AIK, SRK, and associated keys

Both designs rely on a *core root of trust* for measurement, a process that initiates a series of software measurements made in sequence during platform boot, comprising a *boot chain*. In the trusted computing context, the TPM's PCR store measurements of all components in the chain enabling sealing, quotes, and other operations. The core root of trust is either some firmware in the BIOS, or the `AC_INIT` software module present on modern Intel processors (AMD has a similar module).

To simplify the presentation, we describe the functional designs of each alternative in terms of a *use case*, or what the voter sees during interaction. The attestation approach is common to both alternatives, and appears separately.

7.3.1 Image Duplicator

The image duplicator scans the images of all markable positions and prints them in a permuted order onto the receipt. The image duplicator requires no knowledge of the Scantegrity ballot encodings, and no connections to the PCOS scanner–thus, we call this a *stateless* design. For this reason, the image duplicator is the simpler of the two receipt printer approaches.

Functional Design

The image duplicator consists of an image scanner coupled with a printing device. For the purposes of this design, the image duplicator's scanner and printer are an integrated unit.

A simple use case sequence best describes the major design elements of the image duplicator. The use case follows:

1. The voter completes her Scantegrity ballot in the polling booth.

2. The voter presents her marked ballot to the image duplicator.

3. The image duplicator scans the *markable positions* of the ballot—all of the Scantegrity bubbles, whether marked or not—and also scans the online verification number from a *Two-Dimensional QRcode (qrcode)*.

4. The image duplicator creates the receipt by printing: (a) images of the marked positions exactly as scanned onto the receipt, but with their order rearranged; and (b) an AIK *digest* of the online verification number[1]

5. The voter verifies the codes from the filled bubbles printed on the receipt with those on her ballot. She also verifies that the online verification number matches her ballot. If there is a discrepancy, she alerts a poll worker and her marked ballot is revoked, consistent with the precinct's practices and procedures.

The voter verifies the integrity of the image duplicator software using the attestation protocol described in Section 7.3.2.

[1] A digest consists of some plaintext values and a hashed representation of them signed by some key, the AIK in this case.

Details

The critical design details of the image duplicator include how the markable positions–the bubbles–are identified and scanned, and how the bubble images are presented and used.

Markable Positions The image duplicator identifies markable positions within individual contests using x, y offsets from preprinted alignment marks detected on the ballot. The Scantegrity ballot uses dark circles to identify the qrcode and the markable positions to the PCOS scanner, and the image duplicator will reuse this feature.

The image duplicator scans an image of the entire area of each markable position. A suggested resolution for the scan is 150 dots per inch and 8-bit grayscale (256 levels of gray), sufficient for resolution of the revealed Scantegrity codes. The image duplicator does not attempt to determine the filled state of the corresponding bubble, but merely captures the image as marked on the ballot.

Scanned Bubble Printing 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. Images of blank bubbles will appear after images of partially marked and fully marked bubbles, respectively. For each contest, the image duplicator prints a contest indicator, e.g.,contest 1, and the scanned bubble images for that contest, onto the receipt.

Receipt Usage The voter compares the verification code images printed on her receipt with those reflected on her ballot, to make sure that the verification codes are correctly and intelligibly recorded. A potential implementation issue is whether the grayscale scanning and print features can render the verification codes legibly onto the receipt;

having the voter verify her codes at this stage can help determine whether the receipt will be useful to the voter later.

An example receipt is shown in Figure 7.1.

Before presenting the marked sense translator, we break for a moment to show the attestation design, as it is the same for the image duplicator as it is for the marked sense translator.

7.3.2 Receipt Printer Attestation Protocol

Our claim is that the authorized software will print the correct information and will safeguard the voter's privacy by not disclosing her preferences. We ensure this claim by giving the voter a way to confirm that the receipt printer loaded the correct software at boot time. To verify the state, the TPM gives the voter cryptographic evidence of the platform software state, and in this way, the TPM is said to *attest* the state of the platform to the voter. The attestation design uses the TPM to report boot-time measurements of the platform state securely.

The two design alternatives for attestation are *pre-scanning* attestation, in which the voter verifies the software prior to scanning her ballot or entrusting the printer with other private information, and *post-scanning* attestation in which the voter verifies the platform software state after scanning her ballot, using the attestation proof printed on it. While the pre-scanning variation is better for privacy, the post-scanning variation is more practical, and still can detect misconfiguration or rogue software before the voter leaves the polling location.

Attestation protects voter privacy and receipt authenticity–it cannot guarantee election integrity, and therefore is not a substitute for the voter verifying the confirmation codes on

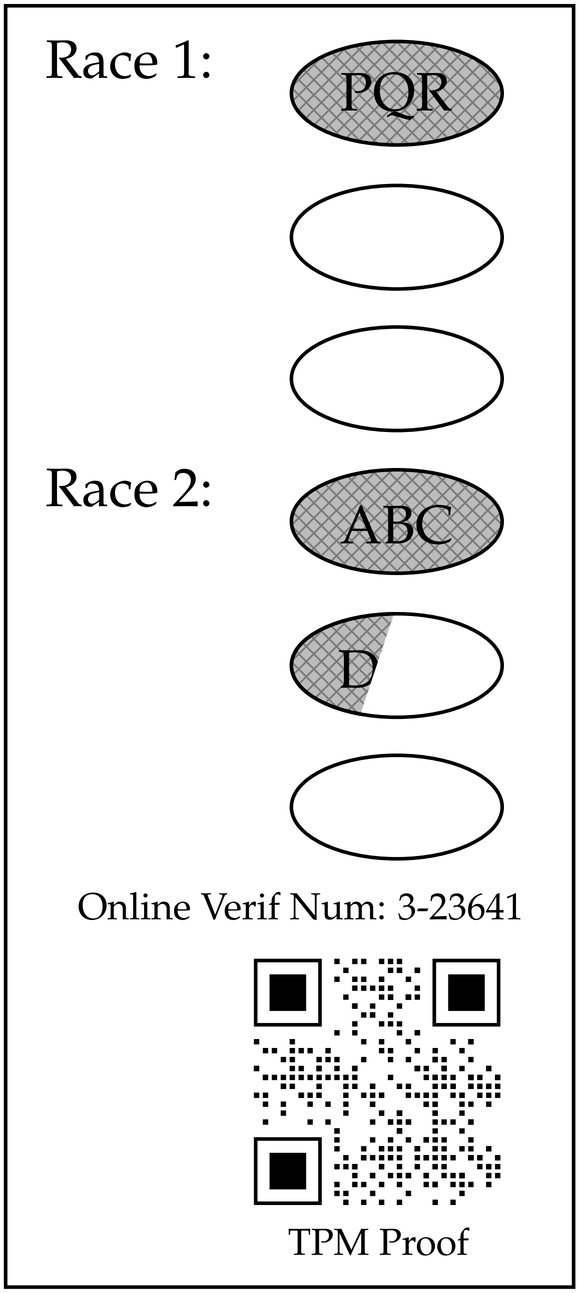


Figure 7.1: Receipt from image duplicator. Images of scanned bubbles are printed in order of average pixel density, grouped by contest. Notice that partial marks also appear. (Overvotes and undervotes are detected by the PCOS later in the voting process.)

her receipt against those on her ballot. The voter, or a trusted third-party voting rights group, should use at least one type of attestation to prove to the voter that she is interacting with a correctly configured receipt printer.

We discuss the individual phases of attestation: system initialization, election day initialization, voter attestation, and termination.

System Initialization

The *Election Authority (EA)* performs some initialization to enable platform attestation:

1. The EA takes *ownership* of the receipt printer TPM. Ownership establishes the key hierarchy in the TPM. It is done only once, and is good for as long as the EA owns the equipment.

2. The EA determines the set of expected PCR values by measuring software components of the approved build for the receipt printer.

3. The EA commands the TPM to create the AIK used for signatures, sets an authorization password on the AIK, and binds the AIK to the expected PCR values. The EA certificate authority signs the AIK public key certifying that it was created by an approved TPM.

4. The TPM creates a monotonic counter bound to the authorization password and PCR values, similar to the AIK. Note, this counter is only useful in the marked sense translator as explained in Section 7.4.

5. The EA creates a delegation of the TPM `ownerClear` command used to destroy the TPM key hierarchy, and sets a special tear-down password on the delegation.

The EA publishes the AIK public key, the AIK certificate, and the PCR values, but keeps the AIK and tear-down passwords a secret until election day. The critical secret component is the AIK private key, and it is exposed only inside of the TPM.

Although several AIK keys should be created per platform to prevent an adversary from correlating a usage pattern to a particular platform, we assume a single AIK to simplify the above narrative.

Election Day Initialization

1. At the start of election day, the EA distributes the AIK password to the poll workers.

2. The poll workers boot the receipt printer. The printer performs a *trusted boot* process where each booted component cryptographically measures each subsequent component, storing the measurements in the TPM's PCRs. The poll workers enter the AIK password into the receipt printer; the password and correct PCR values enable the use of the AIK.

3. The receipt printer platform prints the digest of the monotonic counter value, signed with the AIK.

4. (Poll workers do other general initialization, e.g.,test scans to check scanner and printer connections, alignment integrity, ink and paper levels, etc.)

Election Day Attestation

1. The voter marks her ballot during voting, and takes it to the receipt printer.

2. In ***pre-scanning attestation***, the voter uses a smart card to verify the receipt printer's integrity using the technique discussed in [63]:

   (a) The voter obtains a smart card from a trusted third party.[2]

   (b) The smart card generates a random number called a *nonce*, and sends it with a quote request to the receipt printer.

   (c) The receipt printer requests a `TPM_QUOTE`. The TPM fetches its PCRs and signs them and the nonce with the AIK, producing the quote.

   (d) The receipt printer returns the quote to the smart card, along with the AIK key identifier. These data are called the *attestation proof*.

   (e) The smart card checks its keys dictionary for the AIK. If found, it verifies the PCR values in the quote reply, and validates the signature.

   (f) (Steps 2b to 2e can be repeated several times by allowing the smart card to time the responses to detect a proxy/oracle attack [63].)

   (g) Upon successful verification, the smart card reveals an *attestation secret* to the receipt printer–a word, phrase or number that means the platform software is correct–and the printer reveals this secret to the voter.[3]

   (h) The voter confirms the attestation secret with the trusted third party, and alerts a poll worker if she cannot confirm the secret.

3. The voter releases ballot details and the ballot online verification number to the receipt printer (design-specific, either by direct scanning or by PCOS transmission).

4. The receipt printer issues a `TPM_QUOTE` as in Step 2c, but using the online verification number as the nonce. The TPM responds per Step 2d.

[2] The trusted third party can verify the smart card using a self-signing timing technique, as explained in [63, 71, 134].

[3] The implementation dictates how the secret is revealed: if the printer has a display, it shows the smart card secret on it; if not, it prints the secret on a blank sheet of paper.

5. The receipt printer prints the Scantegrity receipt, including the Scantegrity codes and the online verification number and attestation proof and additional design-specific proof as required. The proof is printed as a qrcode for ease of use.

6. In ***post-scanning attestation***, the voter uses a trusted device to read the qrcode, look up the AIK public key, and verify the attestation proof. The trusted device can be a *Personal Digital Assistant (PDA)* or cell phone, or can be a separate computer maintained by a trusted third party, possibly a voting rights group located in the polling place.

7. If the attestation proof fails verification, the voter alerts a poll worker to take remedial action.

Election Termination

At the end of the election period, the platform signs the final value of the monotonic counter with the AIK. When the EA releases the `ownerClear` password, the poll workers enter it causing the TPM to erase its key hierarchy so that the AIK private key never can be used again. Although not part of attestation, the poll workers can retrieve digital archives of the receipts at poll close time, to publish for verification by third-parties.

Cryptographic Keys Summary

The AIK (and the *Verification Codes Secret (VCS)*, used by the marked sense translator for confidentiality described in Section 7.3.3) have several properties shown in Table 7.1.

We will now show an additional use of the TPM in the second variant of the receipt printer described below.

Operation	**AIK**	**VCS**
Purpose	Prove platform booted correct software	Protect Scantegrity verification codes from disclosure
Type	Asymmetric (public/private)	Asymmetric (public/private)
Owner	EA	EA
Key Creation	At platform initialization	Just after platform initialization
Import/Export	Public/plaintext	Sealed blob
Distribution	One or more unique AIK per platform	One VCS shared over all platforms, wrapped by individual platform SRK
Parent	Internal TPM *Endorsement Key (EK)*	TPM SRK public
Key Use	Sign online verification number and PCR values	Decrypt Scantegrity verification blob
Authorization	PCR values, election day password	PCR values, election day password

Table 7.1: Properties of keys in the Scantegrity receipt printer.

7.3.3 Marked Sense Translator

The marked sense translator connects directly to the PCOS scanner. It receives mark sensed positions from the PCOS, translates the positions into the Scantegrity codes that should be revealed on the ballot, and prints the codes onto a paper receipt. Unlike the image duplicator, the marked 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.

Functional Design

Figure 7.2 gives an overview of the marked sense translator in operation. A use case describes the high level design:

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

2. The PCOS scans the marked bubbles and ballot ID from the ballot. It interprets the marked bubbles as *selections*. The PCOS sends the selections and ballot ID to the marked sense translator. Other data, such as overvote or undervote details compliant with *Help America Vote Act (HAVA)* requirements, may be transmitted also [1].

3. The marked sense translator securely retrieves the Scantegrity verification codes for each selection, and prints them onto a paper receipt. It also prints an AIK digest of both the voter's codes and the monotonic counter value. The TPM increments its counter.

4. The voter verifies that the Scantegrity codes on the receipt match those on her ballot.[4] If there is a discrepancy, she alerts a poll worker and her cast ballot is retrieved and revoked, consistent with the precinct's *Policy and Procedures (P&P)*.

The voter verifies the integrity of the marked sense translator software using the attestation protocol described in Section 7.3.2.

Details

The critical design details of the marked sense translator include contents of the receipt, protection of the Scantegrity verification codes, and modifications required of the PCOS.

Receipt Contents As with the image duplicator, the marked sense translator groups the revealed codes by contest. It randomly orders the codes within each contest. Unlike the

[4] Section 7.3.3 discusses practical ways of comparing codes.

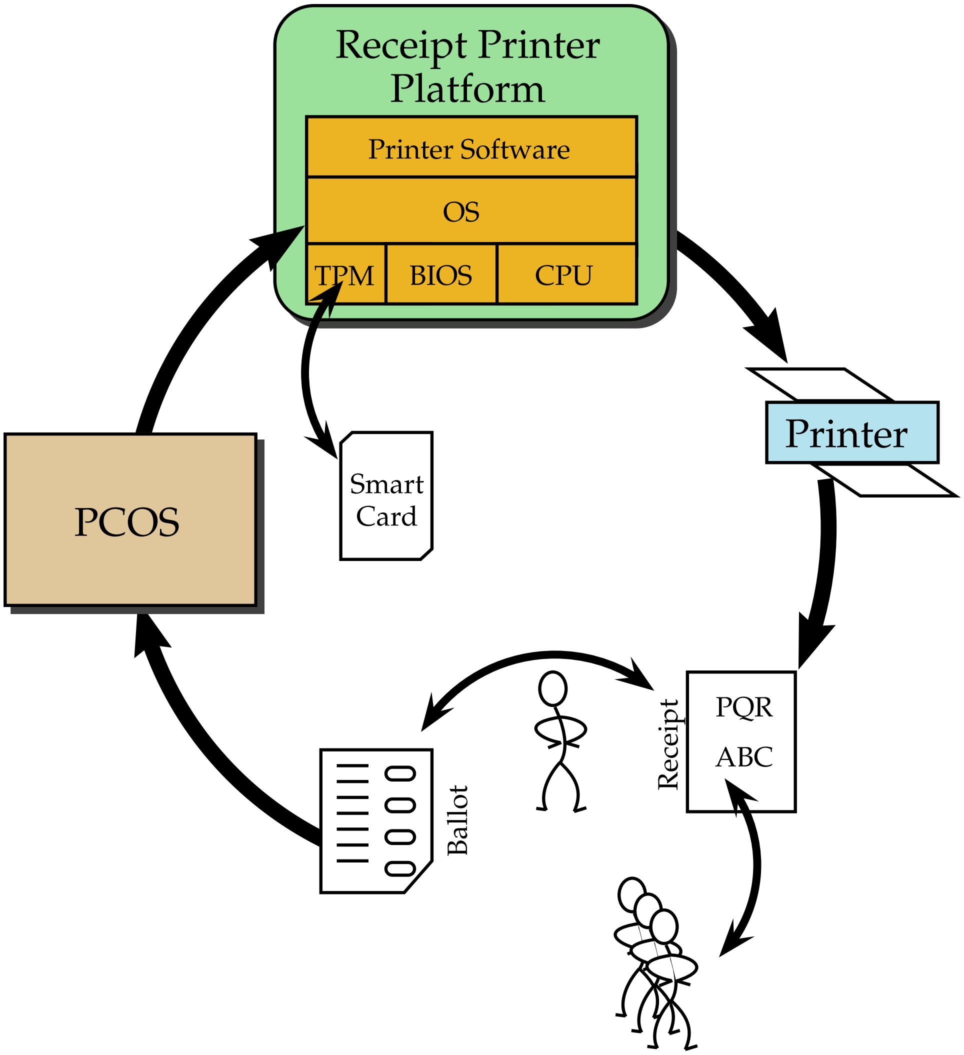


Figure 7.2: Overview of the marked sense translator. The voter submits her ballot to the PCOS that sends the marked positions, optional encrypted ballot definition, and online verification number to the marked sense translator. The voter optionally uses a smart card to verify the platform. The software uses the TPM to reveal the Scantegrity codes, prints the codes and attestation proof onto the receipt. The voter compares the receipt to the ballot. Anyone may verify the integrity of the receipt and the marked sense translator with the attestation proof on the receipt.

image duplicator, the marked sense translator does not scan the ballot, and therefore it does not report codes of partially marked bubbles, those not sensed by the PCOS.

Connection to PCOS The PCOS is connected to the marked sense translator using a data cable. The data sent to the marked sense translator include (a) ballot ID; (b) contest designations; (c) marked positions by contest (e.g.,“contest 1, position 1 of 3 is marked”); and (d) optional indication of overvoting or undervoting per contest.

The PCOS must enforce a well-defined message interface format to protect it from a corrupt marked sense translator that may send ill-formed messages to the PCOS. A one-way data cable *may* mitigate this threat, but it might break any message that requires acknowledgment from the marked sense translator.

Scantegrity Codes Retrieval The marked sense translator uses the private portion of its VCS–bound to the platform PCRs–to decrypt the codes corresponding to the scanned ballot. The codes for each ballot are encrypted uniquely for every platform by the EA, and are indexed by ballot ID. Further, the EA signs the codes blob proving authenticity of the codes to the marked sense translator.

Required PCOS Modifications The PCOS must be modified to supply the ballot ID and sensed marked positions to the image duplicator. These data are part of the content already retained by the PCOS.

Cryptographic Key Summary

The EA creates the VCS and encrypts it with the public portions of the SRK of each platform. A common password restricts the VCS use until election day, and the key is sealed to the receipt printer PCR values. The EA loads the encrypted VCS blobs onto the platforms during system initialization. Table 7.1 in Section 7.3.2 summarizes the properties of the VCS keys.

Design Enhancements

Some enhancements to the basic design include how to display the scanned ballot, and how to transport the Scantegrity codes efficiently and securely to the marked sense translator.

Ballot Image Display

The voter needs a way to cross-check the Scantegrity codes on the receipt with those on her ballot. Unlike the image duplicator, the marked sense translator lacks a scanner and relies on the PCOS for its information. As a consequence, once the voter submits her ballot to the PCOS scanner, it is no longer available to her.

Although a straightforward solution is to have the voter handwrite a few spot check codes from her ballot onto a piece of scratch paper to check against the receipt, this is problematic and defeats the purpose of the printer. Therefore, we suggest two usable design alternatives that let the voter see her votes to ensure that the PCOS sensed the correct marks: a *lever cast* mechanism on the PCOS, and an integrated graphical ballot display.

The lever cast mechanism is a glass display case attached to the output end of the PCOS scanner. The glass case makes the scanned ballot fully visible–but unalterable–until the voter pulls a physical lever to release it into the ballot hopper. After receiving her receipt, the voter compares it to her ballot under the glass, and if the codes match, she pulls the lever to drop the ballot into the hopper; if there is any problem, the ballot can be retrieved and revoked.

The integrated high-resolution display renders a graphical image of the ballot with the Scantegrity codes filled into the marked ovals. It recreates the ballot using information sent from the PCOS. The display shows (a) individual contest and possible choices as printed on the physical ballot; (b) blank bubbles for unselected choices (including undervoted contests); (c) filled bubbles showing Scantegrity codes (including overvoted contests); and (d) the ballot ID number and other information that will be printed on the receipt.

After receiving her receipt, the voter scrolls through the display and ensures that (a) the displayed ballot reflects her intent; and (b) the Scantegrity codes on the receipt match those on the display.

The display could incorporate magnification or an audible interface for improved accessibility.

For voter privacy, the image on the display can never be captured or printed, just as we do not photocopy the Scantegrity ballot itself.

Transporting State with the Ballot

One challenge is transporting the ballot state–the Scantegrity verification codes–to the marked sense translator so that it can print the correct verification codes. A simple design is for the EA to encrypt the verification codes for every ballot with the VCS, and load these onto the platforms' persistent storage prior to election day. Unfortunately, this reduces flexibility by requiring extra work prior to the election.

A smarter design would have the PCOS read the encrypted verification codes from the ballot itself, and transmit these to the marked sense translator. The qrcode printed on the 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.

7.3.4 Policy and Procedures

With both designs, some general procedures must be followed to ensure the security of the system. At a high level,

- All authorized receipt printers must be in visible locations, e.g.,no printer can 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 or the TPM [5]

The security of the system relies on a majority of poll workers knowing, and correctly enforcing these policies and procedures, regardless of design choice.

7.4 Security Analysis

We rely on the assumptions of TPMs for our system security, but in this section we evaluate the total impact a receipt printer will have on election security. We analyze what happens when these assumptions are violated by an attacker to get control of a receipt printer, and what types of attacks he could perform.

7.4.1 Threat Model

We limit our model to attacks that utilize a rogue receipt printer during the election. See Chapter 3 for a more general security analysis of Scantegrity and voting, and Fink *et al.* [62]

[5] Software injection is “fair game” as it is done much more quickly than microprocessor delayering attacks

for an analysis of TPM protocols for voting. In the election context, our adversary could be 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 general 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 certification of the election undetectably.
4. *Discreditation Attacks*, where an attacker imbues sufficient doubt in the public's perception of Scantegrity's worth.

We do not consider denial of service attacks explicitly, although *disruption attacks* are similar. A denial of service attack is applicable to any voting system and is difficult to prevent but easy to detect. Covert disruption attacks can be considered a special case of denial of service, where the adversary undetectably delays certification of the election when it suits his purpose.[6]

Many attacks involve procedural elements that are not easily captured by a cryptographic description, so our intent is not to establish and prove security properties in a formal cryptographic model. Instead, we provide an informal analysis of the underlying security goals in our design. Because our analysis is limited to attacks involving the receipt printer, we consider their designs successful if they do not increase the ability of an adversary to carry out successful attacks undetectably.

[6] Undetectability is central to disruption. While attestation failure of the receipt printer may delay the election, at least the EA knows the problem and the printer unit in question. A successful disruption attack will offer little clue as to where the problem is.

7.4.2 Assumptions

Assumptions listed here are limited to those made on the receipt printer and how it should be used in an election scenario. Some assumptions, such as unreadability of the codes, are required by Scantegrity and not just the receipt printer.

1. TPM Integrity–the TPM correctly implements the TCG specifications and does not leak information it is entrusted with. In particular, the TPM safeguards the AIK private key, the monotonic counter, and the ownership authorization secrets.

2. Supporting Hardware Integrity–the platform BIOS and `AC_INIT` module, if applicable, initiates the measured boot process correctly and stores initial measurements in the TPM's PCRs. The scanner, printer, and integrated display mechanisms operate correctly.

3. Software Correctness–platform software is free of critical bugs or supply chain trap doors, and does not become compromised during runtime.

4. Trusted Actors–(a) EA correctly manages ballot creation and privacy, the *Certificate Authority (CA)*, the public bulletin board, and the AIK and ownerClear secrets; and (b) a majority of poll workers follow correct P&P (further described in Section 7.3.4).

5. Voter Actions–(a) voters check the receipt's online confirmation number prior to releasing the ballot; and (b) for the image duplicator, voters check the Scantegrity confirmation codes in addition to the online number.

6. Ballot Security–(a) obtaining information about a voter's preferences given only the confirmation codes is intractable; (b) scanning equipment cannot read unmarked codes printed in invisible ink; and (c) confirmation numbers and printed mark positions do not contain subliminal information that can influence voters.

We believe that these assumptions are reasonable, standard in the literature, and correspond to existing assumptions about Scantegrity and the TPM as used for voting.

7.4.3 Manipulation Attacks

Manipulation attacks are when the adversary tries to change the outcome of the election without anyone noticing. They succeed in Scantegrity only when not enough voters verify their confirmation codes on the public bulletin board. A malicious receipt printer could cause the voter to verify the *wrong* codes against the bulletin board, if the voter fails to verify all the details of her receipt, enabling an efficient manipulation attack.

For example, a malicious receipt printer could perform a *chain printing* attack where it caches and reprints a valid scanned image and ballot online confirmation number saved from a previous ballot, leading the voter to verify the wrong ballot on the bulletin board. In this way, chain printing would "pigeonhole" Scantegrity verifications preventing certain ballots from ever being verified, while misleading voters into verifying other ballots multiple times. The PCOS could flip votes of the unverified ballots, altering the outcome of the election.

Chain printing is detected using the image duplicator if the voter verifies her online verification number and enough of her confirmation codes before she releases her ballot to the PCOS. In the marked sense translator, a voter or an independent third-party verifies the AIK signature on the receipt and carefully tracks previously seen online verification numbers, posting observed numbers to a public bulletin board. In both the image duplicator and the marked sense translator, if the voter verifies the attestation proof on the receipt, she detects malice in the receipt printer restricting ballot modifications to a malicious PCOS and detecting a discrepancy on the public bulletin board–precisely the attack case that Scantegrity is designed to prevent.

Further, keeping digital archives of the receipts enables independent verification of every receipt very quickly and easily, showing an actual improvement in election integrity by combining Scantegrity with a receipt printer.[7]

In all of these cases, the attack is detectable by any attentive voter that is affected by it. It would not be clear, however, if the attack was malicious or simply an equipment failure, though the equipment could be flagged for inspection and not used for the rest of the voting period.

Other defenses include audited ballot scanning where auditors vote and run test ballots through all receipt printers to check correctness of the receipt generation. An audit like this would provide an advantage in the marked sense translator design, because the auditors would be testing the scanner hardware (not software) at the same time.

Because Scantegrity only counts the ballots that were cast correctly, it is potentially vulnerable to ballot stuffing attacks in the event that officials fail to count the number of voters accurately. A receipt printer cannot prevent election day ballot stuffing because the PCOS, not the printer, is the authoritative vote tallying device. In particular, the image duplicator has no connection to the PCOS and cannot track any part of the process. However, the marked sense translator acts as a trustworthy counter and can act as a backup to official voter counts for detecting polling place stuffing attacks. If the marked sense translator is malicious, the signatures and attestation evidence will fail verification. If the PCOS is malicious but the marked sense translator is good, two scenarios play out that involve the monotonic counter:

[7] When using DREs, many voters do not check the printed receipts when voting [60], and this might be true for the receipt produced by the printer; therefore, automatic checking of digital archives is extremely useful.

1. PCOS fails to communicate to the marked sense translator: the EA observes a discrepancy in the number of printed receipts compared with the number of digital vote records included in the PCOS storage.

2. PCOS communicates to the marked sense translator to print a receipt: adjacent, legitimate voters notice a discontinuity in the counter value stamped on their receipt. Also, poll workers notice a higher receipt print count than the number of actual voters at the end of the day.[8]

   Attackers could produce valid receipts for stuffed ballots in this case, but the discrepancy in voters versus printed receipts would have called all the results of the affected poll into question, including the attackers' stuffed ballots.

As mentioned, the image duplicator cannot help detect election day ballot stuffing attacks because it is not integral to the ballot casting process. Same for the marked sense translator if there are some independent PCOS not connected to one.

7.4.4 Identification Attacks

An identification attack with a malicious receipt printer would allow an adversary to determine how a voter has voted simply by recording the voters choices and the online verification number. Alternatively, to avoid having to be accessed by the adversary later, it could print hard to notice "markers" on the receipts to indicate the selection by the voter, through a subliminal channel in the signature or by some graphical marker on the receipt. It would be difficult to detect or prevent these privacy attacks.

[8] To prevent "behind the curtain" ballot stuffing attacks, poll workers note which authorized machines participated in the election [62]. Additionally, TPM *tick stamps* could stamp the "time" on the receipt for correlation with observed arrival times of authorized voters.

If assumption 6c is violated, an attacker could use the information on the receipt provided by the receipt printer to coerce or identify the selections made on that ballot. An attacker would have to violate this assumption by subverting the printing authority or the Scantegrity trusted workstation. This attack is somewhat orthogonal to the receipt printer, but it could succeed in the image duplicator design because the images on those receipts are not posted publicly. An attack could be discovered in a marked sense translator design because all of the information is publicly posted to a bulletin board, allowing auditors to verify the pseudorandomness of the Scantegrity confirmation numbers.

The image duplicator design may be susceptible to tampering that would reveal ballot selections. If voters can modify the ballot to skew the alignment detection, a slight angle and/or offset may make it possible to determine the order the codes appeared on the ballot. The implementation should be strict regarding the alignment detection and reading the online verification number, and should not print a receipt if it cannot align the ballot image.

7.4.5 Disruption and Discreditation Attacks

The manipulation attacks described in Section 7.4.3 can be used as disruption attacks. In general, simply having misbehaving equipment can affect the perception of trustworthiness in the election, but it does not inherently prevent certification of the election.

An undetected malicious receipt printer could print and sign illegitimate receipts e.g.,chain printing. This would delay certification of results, as the paper record would have to be consulted to determine the legitimacy of false claims.

The marked sense translator design, because it has access to all the confirmation numbers of the scanned ballot, could enable an attacker to submit false challenges for every ballot it sees. But for this to work, assumption 1 must be violated. This attack would certainly disrupt certification of the election, and the electorate might, rightly, believe that the privacy of the election had been compromised upon discovery of such a large number of legitimate complaints, leading to disruption and discreditation of Scantegrity. Another variation would reveal additional codes to voters (as in an overvote). Voters may not notice in the polling place, and could legitimately complain that the receipt does not match the online record after the election.

Receipt forgery is not possible with the marked sense translator, but cannot be prevented easily by the image duplicator. The marked sense translator signs the ballot state (Scantegrity verification codes) with its AIK preventing forgery. Since the attestation evidence is printed on the receipt, anyone can verify the receipt for authenticity and correctness. Unfortunately, the image duplicator cannot prevent forgery, because the only state it has are the scanned representations of the bubble images. Rescanning a physical artifact may lead to different digital values of the page, a fact used to great benefit by certain randomization techniques e.g., [105]. Image processing algorithms may be able to "bin" the scanned ballot images into a small set of discrete values, making it difficult to forge the receipt without violating the signature.

7.5 Discussion

Both receipt printer designs provide two distinct advantages over using the underlying election system only:

1. Usability is improved when voters can use the printers to produce receipts automatically. Automatic receipts save voters time and energy, making each voter more likely to produce and check the receipt after the election. It may decrease the amount of time each voter spends in the polling site and increase the flow of voters through the

polling place. Voters who accidentally undervote or overvote the ballot can also be better informed about what happened via the receipt printer.

2. Security is improved through signed, publicly verifiable receipts. Such receipts weaken several attacks that may exist in the underlying system. A signed and authenticated receipt gives a voter stronger evidence of recording errors than simple knowledge of a code. A receipt that does not authenticate properly is proof of an equipment issue that should be investigated. The receipt printer can also now provide proof of an under- or overvote, preventing a ballot from being invalidated or allowing change of intent.

The main disadvantage of the printer designs is that, in general, they become an attractive target for violating voter privacy during the election and must be implemented carefully. Because they are polling place devices, even under the assumption that the central authority can store them properly, an attacker could have up to several days to tamper with the machines when they are deployed shortly before the election, including carrying out costly physical presence attacks [128].

Assuming the hardware is secure, an attacker could not fake a valid receipt. This makes an attack on the printer costly and unlikely. It is unclear when fake hardware would be noticed, and that could pose a denial of service problem.

Only one voter needs to find a discrepancy for an attack on the printer to be discovered. If the intent of the attacker is to attack privacy or election integrity, then a 50% chance of detection makes it a very high risk approach. If the intent is simply disruption, these attacks would be effective; however, there are many simple and presumably less costly disruption attacks (e.g.,distributed denial of service on the bulletin board mechanism where receipts are posted).

7.5.1 Comparison of the Designs

The image duplicator design is simple compared to the marked sense translator. The marked state of each bubble is ignored, avoiding complex algorithms to decide whether a bubble is filled, blank, or partially filled. Printing the bubbles in order of average pixel intensity, within contests, is straightforward and gives enough position permutation to protect how the voter voted.

The image duplicator design has the advantage of being intuitive to voters. Because it is simple and a completely separate component with one function, it is easier for a voter to understand what it is supposed to do. Also, to a security conscious voter, it may appear like a more secure design decision to deploy it as a separate component. Making it a separate component may allow the receipt printer—a completely new device—fit in better with a voter's mental model of elections ("Oh, this thing is supposed to give me a receipt to improve security, I get it"). Attaching the receipt printing to an existing component with a well known function may make it harder to comprehend what is happening.

As a standalone independent component, the image duplicator could be used as an option and would therefore have minimal impact on voter flow through the polling site. However, the independence could introduce problems with voter flow if confused voters tried to scan their receipt or other materials instead of their ballot.

The marked sense translator design makes the process easier for the voter. It eliminates having to scan the ballot twice. It allows the voter to verify that the PCOS sensed and recorded the ballot marks correctly, and if there is a problem on the ballot (e.g.,overvotes) it can be shown to the voter and poll workers immediately, catching problems early.

The marked sense translator design allows for better accessibility features. Because the selected confirmation numbers are known, they can be provided easily with audio. Since the Scantegrity codes reveal nothing about the voter's preferences, the marked sense translator could also broadcast the confirmation numbers over a standard interface, allowing voter-controlled devices to get signed digital copies of the receipt on their own trusted devices (this is very useful for blind voters). The image duplicator design could be outfitted with a similar interface, but—barring a optical character recognition algorithm—is limited to providing the image data only.

The marked sense translator design, in general, has a smaller digital footprint than the image duplicator design. A 3-digit confirmation number can be expressed in 3 bytes or less, depending on the number of symbols used to create the confirmation number. Images generally require much more space to store. Thus, the marked sense translator design produces smaller receipt files than the image duplicator design. This can turn out to be a advantage to an auditor who needs to verify a large number of receipts.

Both designs add cost to the election. The image duplicator design incorporates scanning hardware that would be more expensive initially than the marked sense translator. Additionally, the marked sense translator requires changes to the PCOS to support the one-way data transfer interface. The long-term cost of operations of both should be minimal, compared to the value of the security and usability benefits.

7.5.2 Design Tradeoffs and Other Considerations

Election officials could deploy both types of receipt printer, as a way of verifying the receipt printers independently from each other, and independently from the PCOS scanner. This could catch problems early, and perhaps a scanner with the marked sense translator attached could be used in addition to other scanners, with only some voters using the marked sense translator for the better accessibility capabilities and reduced overall cost (and reduced likelihood of catching ballot stuffing attacks).

A one-way cable between the PCOS and marked sense translator provides extra protection to safeguard the PCOS from a rogue marked sense translator. There is no other security property, and therefore a two-way cable could be used as long as the PCOS implements a rigorous interface definition. Enabling the PCOS to hang onto ballots after scanning helps the poll workers to retrieve the ballot easily in the event of an attestation failure or a receipt check problem. It is important that the cast lever be mechanical in this situation (even if the mechanical lever merely moves a computer controlled motor into contact with the ballot) because the scanner should not be able to make the decision to cast without action from the voter. Otherwise it would not be possible to determine if a complaint from a voter is legitimate.

In the marked sense translator, the confirmation codes decryption key cannot be used if the marked sense translator platform booted the wrong software, but transporting the encrypted confirmation numbers on the ballot is better for security because not all codes would be known to all TPM. This limits exposure in the event that an adversary gains control of a TPM. Using an AIK instead of any other key keeps knowledge of the *specific* TPM a secret, while proving that the signature came from *some* valid TPM, thwarting voter identity attacks.

7.6 Extensions

The marked sense translator, described in Section 7.3.3, produces verifiable codes to the voter in a highly readable format—including audible formats, not possible with the image duplicator—and it catches any marked position sensor errors in the polling place (or vote flips!) committed by the PCOS. These features make the marked sense translator the best design for usability, accessibility, and security, and is the best way to employ a printer in the Scantegrity architecture.

7.6.1 A Design without Invisible Ink

Since we entrust the marked sense translator platform with the Scantegrity confirmation codes, can we just print the codes on the receipt instead of printing them redundantly on both the receipt **and** the ballot? If we could, we can eliminate invisible ink altogether.

The basic design prints the codes on a receipt, becoming the way that the voter receives her Scantegrity confirmation codes. A key design consideration is that if we *late-bind* the codes to the ballot using the receipt, we must enable the voter to verify that the codes actually correspond to her choices on her ballot—that is, she must be able to tell that the PCOS sensed her marks correctly. In the marked sense translator, the voter submits her ballot to the PCOS and no longer has access to her ballot, except to look at it behind glass in a lever cast configuration.[9] There are two cases to consider.

Random Order Ballots

Some jurisdictions allow random candidate reordering on the ballots, such that the order of candidates within each contest varies from ballot to ballot (like it does with Punchscan [119]).[10] In the random reordered ballot case, the marked sense translator could print a copy of the ballot, omitting candidate names but printing the confirmation codes in the correct position, or listing an index of marked positions and corresponding codes. The voter confirms that the marked positions are correct before casting. This approach resembles Chaum's visual cryptography idea, described in [34].

[9] The image duplicator is a stand-alone design, and therefore cannot verify the PCOS.

[10] This avoids the *primacy* problem, where undecided voters choose the first listed candidate more than any other. See Miller [97].

For example, consider three choices in a hypothetical race for president, Bush, Buchanon, and Gore, in that order, with the voter selecting Buchanon on the ballot. The PCOS scans the ballot and transmits the selection to the marked sense translator, which prints the confirmation code for Buchanon plus an index of "middle selection made." The voter checks her ballot to ensure that Buchanon is the middle choice in the race. The voter checks the election bulletin board later for her Buchanon code.

Fixed Order Ballots

Some jurisdictions object to random ordering, arguing that massively duplicated, fixed-order ballots are less confusing to voters and are cheaper to produce than randomly ordered ballots. In the fixed-order ballot case, the marked sense translator can detect PCOS errors only if it can reveal the voter's selections safely, without violating her privacy. To do this, it creates a temporary replica of the voter's ballot, and either displays it on a graphical (or audible) terminal, or prints it on a separate sheet that must be destroyed before the voter leaves the scanning station. The voter checks the replica of her ballot, and ensures that it reflects her intent before releasing her ballot for casting.

7.6.2 Concerns

Critics may argue that a replica ballot, used in the fixed-ordered ballot case, threatens voter privacy. Researchers including Sherman *et al.*studied vote verification devices, and concluded that voter privacy is at risk with many of these systems [137]. Since the marked sense translator is a vote verification technology, similar problems could exist. For instance, the voter verifies the replica ballot in the scanning booth, and the poll worker potentially could see the ballot and learn the voter's preferences.

A display covered by a special hood could mitigate the poll worker threat, but curiously, this threat exists already in the lever cast mechanism—the same security controls for lever cast should be applied to the replica ballot.

By trusting the marked sense translator as the place where the voter first sees her codes, we could eliminate invisible ink and all the problems that go with it. However, the most significant concern is that the voter may no longer easily vet the correctness of the receipt without trusting the receipt printer or a parallel-testing procedure.

Removing the invisible ink enables officials to deploy traditional optical scan PCOS ballots that are cheap and easy to produce while still being able to catch errant or malicious PCOS behavior as soon as it happens. Voters might require less assistance with a traditional ballot than an invisible ink one, reducing the burden on officials while preserving voter privacy. Additionally, trusting the digital print subsystem with the codes makes it easy to add audible accessibility devices, removing the need for custom Scantegrity ballot markers and verifiers. By late binding the codes to the ballot, we also eliminate chain of custody attacks against the ballot. Of more concern is that the prototype form of invisible ink darkens over time revealing the codes—and revealing the inherent risks of protecting privacy through chemistry and material processes, risks that are much less with traditional printing processes. In short, the marked sense translator can revolutionize Scantegrity by enabling alternative interfaces and eliminating the most costly and problematic part of preparing the ballots.

Chapter 8

Conclusions

Traditional optical scan voting systems have the clear benefit that “votes are verifiably cast as intended”—the voter can see for herself that the ballot is correctly filled out. Yet once her ballot is cast, the voter must place her trust in others that ballots are safely collected and correctly counted. With end-to-end voting systems these last two operations (collecting ballots and counting them) are verifiable as well: voters can verify—using their receipt and a website—that their ballot is safely collected with the others, and anyone can use the website data to verify that the ballots have been correctly counted. The Scantegrity voting system provides such end-to-end verification capability as an overlay on top of traditional optical scan technology.

8.1 Summary and Discussion

There are several interesting observations between the Mock election and municipal election studies. There are also some notable conclusions to be made regarding the receipt printer designs.

8.1.1 Mock and Municipal Elections

The mock election demonstrated that Scantegrity can be effectively used in elections and is well accepted by voters, and the successful use of the Scantegrity voting system in the Takoma Park election of November 3, 2009 demonstrates that both voters and election officials can use sophisticated cryptographic techniques to organize a transparent secret ballot election with a familiar voting experience. The election survey results showed that voters and election judges feel comfortable with the system and have confidence in it, indicating that end-to-end voting technology has matured to the point of being ready and usable for real binding governmental elections. This dissertation thus documents a significant step forward in the security and integrity of voting systems as used in practice.

The mock revealed that the flow of people through the voting process had to be greatly improved, that the locked clipboard added complexity but did not enhance security, and that revealing ink provides a superior technology for marking optical scan ballots with perfectly darkened ovals. The implementation, procedures, voter instructions, and equipment of Scantegrity used in an election need to be simplified and streamlined. Although Scantegrity significantly simplifies the voting process from its predecessors SureVote and Punchscan, additional attention is needed to improve and fine tune the voter experience, including the physical processes of receiving, marking, and scanning the paper ballot. Many of these issues were fixed during the municipal election, where time to vote decreased considerably from approximately 8 minutes per voter on average to under 3 minutes.

During the mock, 31 of the 95 voters verified their votes on line, demonstrating that a sufficient number of voters will likely take advantage of the verification option in E2E systems. This percent of voters verifying their votes was consistent with that observed in other Punchscan and Scantegrity trials. We conjecture, however, that in binding elections, the percentage will also depend on the degree of interest in and contention of the races. The actual receipt check rate during the municipal election was a meager 4%, which was still sufficient but substantially less than during the trials. While we believe this number could be improved with better voter education, it still illustrates the value of performing tests in actual election environments as opposed to trials.

The findings in the municipal study provide further support that voters react positively to the system, that they value the security provided even if they did not care to use it, that the extra work of optionally noting down confirmation codes did not substantially negatively impact the voter experience, and that voters accepted the system in spite of not understanding the inner workings of the system completely. Voter comments indicated that there had been some confusion about specific aspects of the system—such as the double-sided pen used to mark the ballot and the privacy sleeve–and that voters would value additional instruction on using the voting system. Election judge responses indicated that the system was not too hard to administer. These responses also indicated that election judges viewed more problems with the system, providing important recommendations for future use. Except for (a) the fact that women tended to agree more strongly than men on the Likert-scale questions, and (b) a correlation between ethnicity and the intention to verify online, we did not observe significant correlation between demographic data and responses. We did observe a correlation among most of the questions, however, leading to the conclusion that most voters were highly satisfied with the system.

8.1.2 The Receipt Printer Designs

Both of our receipt printer designs have advantages and disadvantages in comparison with each other. While the image duplicator design has some advantages over the marked sense translator design due to its simplicity, the lack of accessibility features and its additional costs are something that should be considered when procuring the receipt printer. The security of the marked sense translator design, while more complex, is not adversely affected by being closely coupled with the PCOS, and the advantages it offers in accessibility appear to outweigh its disadvantages. Because they are independent and offer complementary benefits, both designs could be deployed.

Three key design decisions greatly improve the security properties over other design choices. First, the attestation function allows any voter, election official, or observer to verify the integrity of the software running on the printer. Second, carefully limiting exposure of secret information by using only data printed on the ballot in each design—instead of giving each receipt printer access to all ballot codes—greatly reduces the privacy risk involved when using a receipt printer. Last, any attack on outcome integrity by producing false receipts are difficult because the voter verifies the generated receipt against the voted ballot in each design. As there is no way to cheat undetectably, it is highly likely such attacks using the printer will be caught.

While the marked sense translator and the image duplicator design have different advantages and disadvantages, both can provide a usable and viable way to generate receipts in the Scantegrity election system automatically. Using the TPM as a trusted base helps us verify that the platforms are using the correct software, the receipts are genuine, and voter privacy is maintained.

The marked sense translator, however, produces verifiable codes to the voter in a highly readable format—including audible formats, not possible with the image duplicator—and it catches any marked position sensor errors in the polling place (or vote flips!) committed by the PCOS. These features make the marked sense translator the best design for usability, accessibility, and security, and is the best way to employ a printer in the Scantegrity architecture in our opinion.

8.2 Limitations of this Work

Scantegrity, being a detection system, is necessarily limited. It is also important to consider if this technology introduces as many problems as it solves as might other verification technologies.

Like all verification technologies, Scantegrity does nothing to authenticate voters. It is not possible to fully prevent ballot stuffing from invalid voters, but, unlike other verification technology, Scantegrity does allow detection of modified, replaced, or removed ballots. It also protects against insider threats to the integrity of the election, as no group of election officials can undetectably change election outcomes, but it does not prevent privacy attacks in the face of corrupt insiders.

Scantegrity does not provide a recovery mechanism. We provide no set process for determining if the Scantegrity, machine, or hand count are ultimately correct. Scantegrity can, however, point to where the investigations for malfeasance should be conducted, as voters who complain can indicate the polling site and perhaps even the machine that may have malfunctioned. While a recovery mechanism would be useful, we note that other verification systems do not have recovery mechanisms, and they also do not often help determine where problems might originate. Scantegrity does offer an advantage when determining the source of a problem due to the forensic fingerprint it leaves on each ballot.

Scantegrity can introduce additional vulnerabilities. The fingerprints on each ballot could allow attackers to violate voter privacy more easily. Also, due to the multiple counts—a machine count, a possible hand count, and the Scantegrity count—an attacker could throw doubt on an election by violating any one of these counts. Note, however, that this problem exists in the underlying optical scan system. Scantegrity does not make the problem worse. This is an underlying theme in much of the Scantegrity design. It does not necessarily protect a voter from being forced to chain vote; it does not change the information on the ballot in a fundamental way that makes the privacy substantially worse; it does nothing to count unclear marks over the underlying optical scan system. Thus, while Scantegrity is not a panacea, it does not appear to have a negative impact on the security or privacy of optical scan elections.

Regarding usability, while we did not observe a negative impact on voter satisfaction, there is one substantial limitation to our study: the ballot was short. It is possible that a longer ballot would negatively impact voter satisfaction.

8.3 Open Problems

Further development should improve scalability (esp. printing), usability (e.g. with printed receipts) and accessibility of the Scantegrity system. In particular, accessibility for voters with disabilities was not a focus of our studies. In separate projects, our team is seeking better solutions for the vital challenge of making high-integrity voting truly accessible to differently-abled voters, including the blind.

For future work, a clear measure of the confidence increase (or decrease) the receipt provides is necessary. A comparison study between Scantegrity and a commercial optical scanning system is a possible next step. Another area to explore is whether enough voters will always use the receipts, which is where a receipt printer could provide the biggest boost to voter participation in this part of the process.

Another consideration that should be studied is if the invisible ink in the Scantegrity system is necessary. By trusting the marked sense translator as the place where the voter first sees her codes, we could eliminate invisible ink and all the problems that go with it. We would enable officials to deploy traditional optical scan PCOS ballots that are cheap and easy to produce while still being able to catch errant or malicious PCOS behavior as soon as it happens. Voters might require less assistance with a traditional ballot than an invisible ink one, reducing the burden on officials while preserving voter privacy but reducing mark advantages gained when using the invisible ink to prevent stray marks.

Additionally, trusting the digital print subsystem with the confirmation numbers could make it easy to add audible accessibility devices, removing the need for custom Scantegrity ballot markers and verifiers. By late binding the codes to the ballot, we also eliminate chain of custody attacks against the ballot. In short, the marked sense translator can revolutionize Scantegrity by enabling alternative interfaces and eliminating a costly and problematic part of preparing the ballots.

8.4 Final Thoughts

We have shown that it is possible to make viable E2E voting systems for modern elections. The design of Scantegrity should scale to elections of any size, although there may be additional issues that occur.

While voters appreciate the ability to verify, the key consideration that will make the system workable and well accepted is that every aspect of the system work well. In our studies, significant attention was paid to small details, such as the ballot layout. Instructions for IRV and how to record a receipt were checked carefully for clarity. Takoma Park used an official staff member to create translations of these instructions. The scanner recognized ballots being placed into the system with a beep. These seemingly unimportant issues led to a reasonably smooth election, and voters subsequently reacted positively to the system.

The fact that election officials had more mixed reactions to the system is important. These individuals saw parts of the system that were not polished. A polished, professional implementation would yield a more positive response from election judges.

With Scantegrity, we have demonstrated a simple and effective way to dramatically increase the transparency and security of elections that use optical scan voting systems. It is our hope that its adoption will help prevent the manipulation of election outcomes, and that it may lead to renewed confidence and participation in democracy. Although many voters do not care much about security and tend to trust voting systems, a small and vocal group of political activists is very concerned about this issue. Deploying systems like Scantegrity fundamentally enhances outcome integrity and directly addresses those activists concerns.

Appendix A

Additional Survey Analysis from the Municipal Election

We conducted additional analysis that is not reported in Chapter 6.

A.1 Relationships between Demographic Data

As part of our analysis, we found several correlations between the demographic data. These may assist in explaining our results. Bubble charts of the major

Age and Computer Use. We found that age was negatively correlated with computer usage ($\rho = -0.5$, $p = 4.4E - 16$, $n = 233$). Figure A.1(a) shows age plotted against computer use in a bubble chart. The chart indicates that computer usage fans out in older respondents, but is high and universal among younger respondents.

Age and Income. Age negatively correlated with income (ρ = -.4, p = 3.6E-9, n = 202). Figure A.1(b) clearly shows a possible anomaly in the sample as the 2 youngest groups report unusually high income. We expected income to go up with age and decline after retirement age.

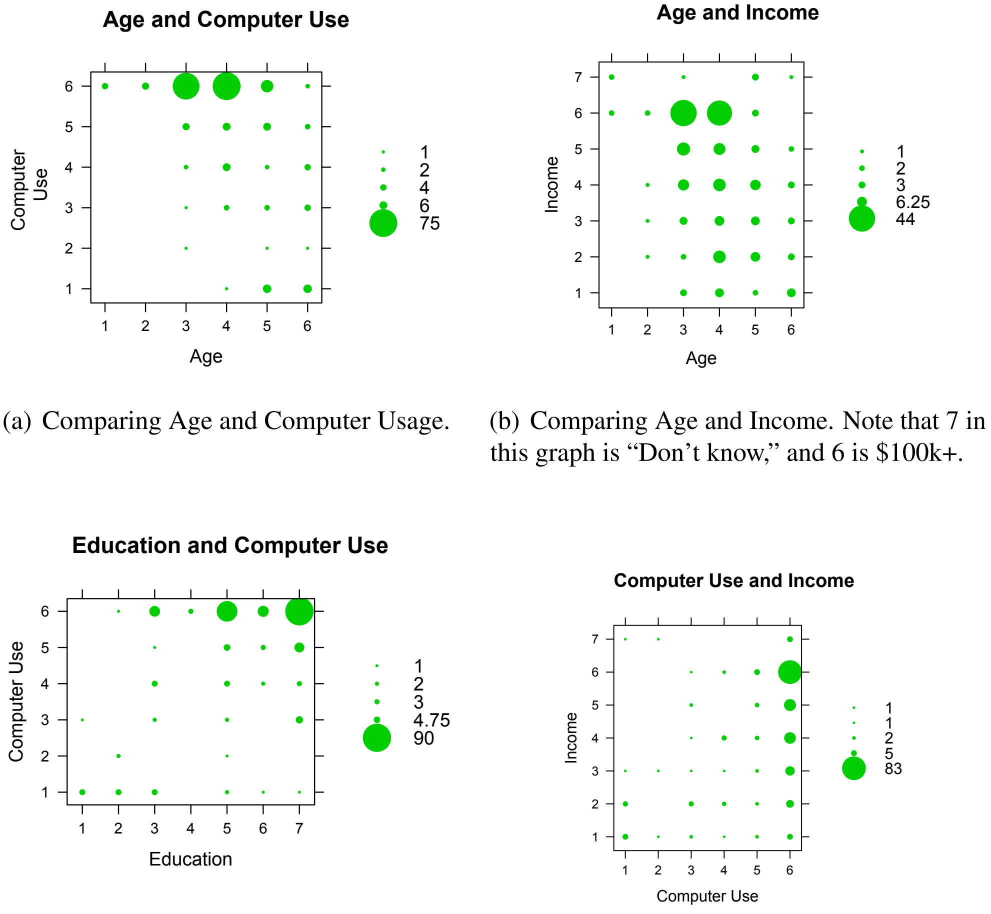
Education and Computer Use. Education positively correlated with Computer Use (ρ = .3, p = 3.16E-6, n = 232). Figure A.1(c) shows a small number of respondents who do not use the computer regardless of education.

Computer Use and Income. Computer Use positively correlated with Income (ρ = .47, p = 3.1E-12, n = 202). Figure A.1(d) shows few high income earners with low computer usage rates.

Race/Ethnicity. Due to low frequency of some categories, race and ethnicity appeared to correlate with every other demographic measure except DQ8 (PrevMistakes) and DQ9 (PollWorker). We combined Hispanic, Multi, and Black (traditionally underrepresented groups) categories and tested our group against Whites. The results still correlated with age, education, computer use, and income.

Other. We found a few other weak correlations. Education correlated with Income (ρ = .28, p = 5.3E-5, n = 202). Voting in previous elections correlated with education (ρ = .15, p = .022, n = 234), Computer use (ρ = .17, p = .01, n = 233), and Income (ρ = .14, p = .04, n = 203).

A.2 Additional Data Tables



(a) Comparing Age and Computer Usage.

(b) Comparing Age and Income. Note that 7 in this graph is "Don't know," and 6 is $100k+.

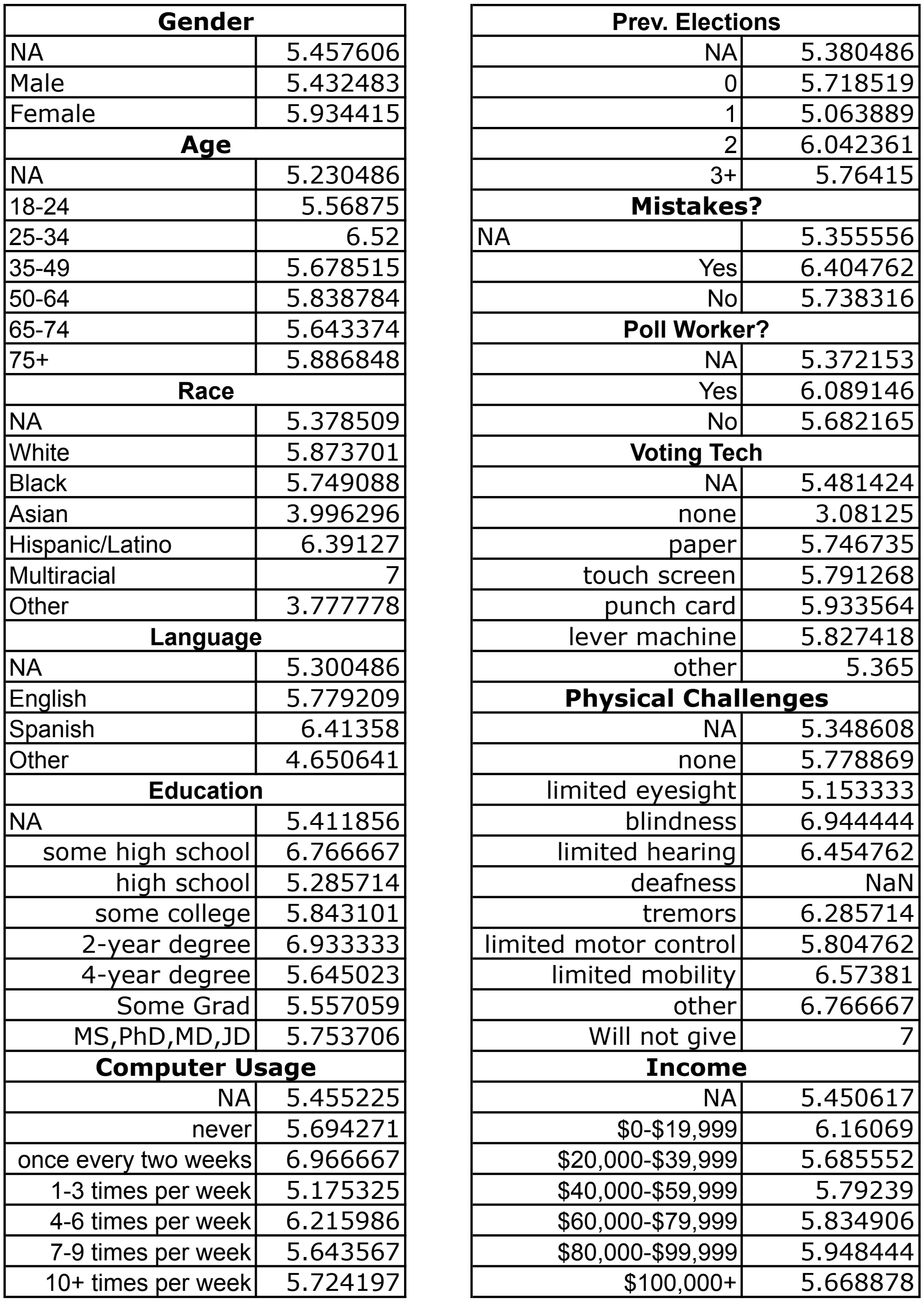
(c) Comparing Education and Computer Usage.

(d) Comparing Computer Usage and Income.

Figure A.1: Different correlations among the demographic data.

Nominal	Q1	Q2	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13
DQ1 (Gender)	10.14	1.20	4.18	7.64	5.00	9.25	0.01	1.92	2.09	0.05	0.49	2.82
DQ8 (Mistakes?)	0.88	1.84	0.33	0.92	1.66	0.36	0.87	0.60	2.06	1.83	2.27	1.54
DQ9 (PollWorker?)	1.72	0.52	3.74	2.90	2.14	1.97	0.02	0.34	0.00	1.91	1.77	0.49
DQ3 (Race)	13.37	6.01	13.24	20.55	15.63	8.90	6.68	6.73	11.52	15.21	6.01	16.92
DQ3 Reduced	0.45	2.36	1.85	0.02	0.03	0.04	0.35	0.25	2.90	3.82	7.40	0.06
P Values												
DQ1 (Gender)	0.00	0.27	0.04	0.01	0.03	0.00	0.94	0.17	0.15	0.82	0.49	0.09
DQ8 (Mistakes?)	0.35	0.17	0.56	0.34	0.20	0.55	0.35	0.44	0.15	0.18	0.13	0.21
DQ9 (PollWorker?)	0.19	0.47	0.05	0.09	0.14	0.16	0.90	0.56	0.97	0.17	0.18	0.48
DQ3 (Race)	0.04	0.42	0.04	0.00	0.02	0.18	0.35	0.35	0.07	0.02	0.42	0.01
DQ3 Reduced	0.50	0.12	0.17	0.88	0.87	0.84	0.55	0.62	0.09	0.05	0.01	0.80
DF Values												
DQ1 (Gender)	1	1	1	1	1	1	1	1	1	1	1	1
DQ8 (Mistakes?)	1	1	1	1	1	1	1	1	1	1	1	1
DQ9 (PollWorker?)	1	1	1	1	1	1	1	1	1	1	1	1
DQ3 (Race)	6	6	6	6	6	6	6	6	6	6	6	6
DQ3 Reduced	1	1	1	1	1	1	1	1	1	1	1	1
Ordinal												
DQ2 (age)	0.00	0.14	0.13	0.13	0.12	0.09	0.03	0.00	0.07	0.14	0.09	0.01
DQ5 (education)	-0.07	-0.19	-0.09	-0.03	0.00	0.07	-0.01	0.05	-0.06	-0.09	-0.12	-0.04
DQ6 (ComputerUse)	-0.06	-0.21	-0.08	-0.14	-0.10	-0.06	-0.11	0.05	-0.08	-0.16	-0.12	-0.02
DQ7 (PrevElections)	0.03	0.15	0.00	0.07	0.07	0.02	0.10	0.03	0.05	0.02	0.00	0.00
DQ12 (Income)	-0.06	-0.21	-0.09	-0.10	-0.08	-0.13	-0.09	-0.01	-0.17	-0.13	-0.27	-0.02
P Values												
DQ2 (age)	0.98	0.23	0.06	0.06	0.07	0.20	0.68	0.96	0.29	0.06	0.25	0.93
DQ5 (education)	0.30	0.10	0.17	0.68	0.95	0.33	0.95	0.48	0.36	0.20	0.12	0.60
DQ6 (ComputerUse)	0.40	0.08	0.27	0.03	0.12	0.39	0.17	0.47	0.29	0.03	0.10	0.76
DQ7 (PrevElections)	0.68	0.21	0.95	0.28	0.32	0.78	0.18	0.66	0.48	0.77	0.96	0.99
DQ12 (Income)	0.39	0.11	0.23	0.14	0.29	0.07	0.29	0.94	0.02	0.11	0.00	0.76
N Values												
DQ2 (age)	230	72	219	229	229	222	171	186	200	186	180	219
DQ5 (education)	229	72	217	228	228	221	171	187	199	185	178	218
DQ6 (ComputerUse)	228	69	216	227	227	220	169	184	197	183	177	217
DQ7 (PrevElections)	230	72	218	229	229	222	171	186	199	185	179	220
DQ12 (Income)	199	59	189	197	198	193	146	161	173	161	158	189

Figure A.2: Demographics correlation tables.



Gender	
NA	5.457606
Male	5.432483
Female	5.934415
Age	
NA	5.230486
18-24	5.56875
25-34	6.52
35-49	5.678515
50-64	5.838784
65-74	5.643374
75+	5.886848
Race	
NA	5.378509
White	5.873701
Black	5.749088
Asian	3.996296
Hispanic/Latino	6.39127
Multiracial	7
Other	3.777778
Language	
NA	5.300486
English	5.779209
Spanish	6.41358
Other	4.650641
Education	
NA	5.411856
some high school	6.766667
high school	5.285714
some college	5.843101
2-year degree	6.933333
4-year degree	5.645023
Some Grad	5.557059
MS,PhD,MD,JD	5.753706
Computer Usage	
NA	5.455225
never	5.694271
once every two weeks	6.966667
1-3 times per week	5.175325
4-6 times per week	6.215986
7-9 times per week	5.643567
10+ times per week	5.724197

Prev. Elections	
NA	5.380486
0	5.718519
1	5.063889
2	6.042361
3+	5.76415
Mistakes?	
NA	5.355556
Yes	6.404762
No	5.738316
Poll Worker?	
NA	5.372153
Yes	6.089146
No	5.682165
Voting Tech	
NA	5.481424
none	3.08125
paper	5.746735
touch screen	5.791268
punch card	5.933564
lever machine	5.827418
other	5.365
Physical Challenges	
NA	5.348608
none	5.778869
limited eyesight	5.153333
blindness	6.944444
limited hearing	6.454762
deafness	NaN
tremors	6.285714
limited motor control	5.804762
limited mobility	6.57381
other	6.766667
Will not give	7
Income	
NA	5.450617
$0-$19,999	6.16069
$20,000-$39,999	5.685552
$40,000-$59,999	5.79239
$60,000-$79,999	5.834906
$80,000-$99,999	5.948444
$100,000+	5.668878

Figure A.3: Mean satisfaction values for each demographic factor. The satisfaction factor is a combination of responses on questionnaire questions 1, 4, 5, 6, 7, 8, 9, 10, 11, and 13.

Bibliography

[1] 107TH UNITED STATES CONGRESS. Help America Vote Act of 2002. http:/www.fec.gov/hava/, 2002.

[2] ADI, A., ADI, W., SÜLER, M., AND FRÖHLICH, P. Demo of ”Open Counting”: A Paper-Assisted Voting System with OMR-Based Public Counting. In *Online Proceedings of the First University Voting Systems Competition (VoComp)* (2007).

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

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

[5] ADIDA, B., DEMARNEFFE, O., PEREIRA, O., AND QUISQUATER, J.-J. Electing a University President using Open-Audit Voting: Analysis of real-world use of Helios. In *EVT/WOTE 2009, Proceedings of the Electronic Voting Technology Workshop / Workshop on Trustworthey Elections, August 10th 2009, Montreal,Canada.* (2009).

[6] ADIDA, B., AND RIVEST, R. L. Scratch & vote: self-contained paper-based cryptographic voting. In *WPES ’06: Proceedings of the 5th ACM workshop on Privacy in electronic society* (2006), pp. 29–40.

[7] APPEL, A. W. Effective Audit Policy for Voter-Verified Paper Ballots. In *Annual Meeting of the American Political Science Association* (Chicago, IL, USA, September 2007).

[8] ARAUJO, R., CUSTODIO, R. F., AND VAN DE GRAAF, J. A verifiable voting protocol based on farnel. In *Proceedings of the 2007 IAVoSS Workshop on Trustworthy Elections* (2006).

[9] ARBAUGH, W. A. The real risk of digital voting? *Computer 37*, 12 (2004), 124–125.

[10] BEDERSON, B. B., LEE, B., SHERMAN, R. M., HERRNSON, P. S., AND NIEMI, R. G. Electronic voting system usability issues. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2003), ACM, pp. 145–152.

[11] 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.

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

[13] BENALOH, J., JONES, D., KELSEY, J., LAZARUS, E., RIVEST, R., AND MILLER, P. Vocomp judges evaluation. http:/vocomp.org/evaluation.php, April 2008.

[14] BENALOH, J., AND TUINSTRA, D. Receipt-free secret-ballot elections (extended abstract). In *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing* (New York, NY, USA, 1994), ACM, pp. 544–553.

[15] BENSEL, R. F. *The American Ballot Box in the Mid-Nineteenth Century*. Cambridge University Press, New York, NY, USA, 2004.

[16] BISMARK, D., HEATHER, J., PEEL, R. M. A., SCHNEIDER, S., XIA, Z., AND RYAN, P. Y. A. Experiences Gained from the first Prêt à Voter Implementation. In *First International Workshop on Requirements Engineering for E-Voting Systems (Re-Vote'09)* (Atlanta, Georgia, USA, 2009), IEEE.

[17] BLACKBURN, R. Appendices to Minutes of Evidence (Volume II): Memorandum by Professor Robert Blackburn, BA, MSc, PhD, FRHistS, Solicitor, Professor of Constitutional Law, University of London (King's College), October 1998. Talks about how the counterfoil practice is still in use in britain just as it was after the 1872 ballot reform act.

[18] BOHLI, J. M., MUELLER-QUADE, J., AND ROEHRICH, S. Bingo voting: Secure and coercion-free voting using a trusted random number generator. In *E-Voting and Identity* (2007), vol. LNCS 4896, pp. 111–124.

[19] BOWEN, D. California Secretary of State Top to Bottom Review. http:/www.sos.ca.gov/elections/elections_vsr.htm, August 2007.

[20] BUECHLER, J., EARNET, T., AND SMITH, B. Voting System Usability: Optical Scan, Zoomable, Punchscan. A student project supervised by Alan Sherman, May 2007.

[21] BURY, J. B. *A history of Greece to the death of Alexander the Great.* Macmillan and Co., Limited, New York, NY, USA, 1913.

[22] BYRNE, M. D., GREENE, K., AND EVERETT, S. P. Usability of voting systems: Baseline data for paper, punch cards, and lever machines. In *In Human Factors in Computing Systems: Proceedings of CHI 2007* (2007), ACM, pp. 171–180.

[23] CARBACK, R. Security Innovations in the Punchscan Voting System. Master's thesis, University of Maryland, Baltimore County, Baltimore, MD, USA, December 2008.

[24] 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.

[25] 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.

[26] CARBACK, R., CLARK, J., ESSEX, A., AND POPOVENIUC, S. On the Independent Verification of a Punchscan Election. In *Online Proceedings of the First University Voting Systems Competition (VoComp)* (2007).

[27] CARBACK, R. T. Printing secure automatic receipts with activating ink, September 2009. http://scantegrity.org/ carback1/ink/ink.pdf; last accessed 25 September, 2009.

[28] CARPENTER, W. *The people's book; comprising their chartered rights and practical wrongs.* s.n., 1831.

[29] CELESTE, R., THORNBURGH, D., AND LIN, H., Eds. *Asking the Right Questions About Electronic Voting*. The National Academies Press, Washington, DC, USA, 2005. http://www.nap.edu/catalog.php?record_id=11449.

[30] CELL, R. I. S. Trusted agent reportdiebold accuvote-ts voting system (raba). http:/www.raba.com/press/TA_Report_AccuVote.pdf, January 20 2004. Technical Report.

[31] 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.

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

[33] CHAUM, D. Elections with Unconditionally-Secret Ballots and Disruption Equivalent to Breaking RSA. In *Advances in Cryptology — EUROCRYPT '88*, D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and C. Günther, Eds., vol. 330. Springer Berlin / Heidelberg, 1988, pp. 177–182. 10.1007/3-540-45961-8_15.

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

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

[36] CHAUM, D. Recent results in electronic voting. In *Presentation at Frontiers in Electronic Elections (FEE 2005)* (Milan, Italy, September 2005), ECRYPT and ESORICS.

[37] CHAUM, D. Scantegrity. In *Online Proceedings of the First University Voting Systems Competition (VoComp)* (2007).

[38] 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 *Proceedings of the 2008 USENIX/ACCURATE Electronic Voting Technology Workshop* (2008).

[39] 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 *Proceedings of the 2008 USENIX/ACCURATE Electronic Voting Technology Workshop* (2008), pp. 1–13.

[40] 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 4* (December 2009), 611–627.

[41] CHAUM, D., ESSEX, A., CARBACK, R., CLARK, J., POPOVENIUC, S., SHERMAN, A. T., AND VORA, P. Scantegrity: End-to-End Voter Verifiable Optical-Scan Voting. *IEEE Security and Privacy Magazine 6*, 3 (May/June 2008), 40–46.

[42] CHAUM, D., HOSP, B., POPOVENIUC, S., AND VORA, P. L. Accessible Voter Verifiability. *Cryptologia 33*, 3 (2009), 283–291.

[43] 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 2005.

[44] Chaum, D., van de Graaf, J., Ryan, P. Y. A., and Vora, P. L. Secret ballot elections with unconditional integrity. Tech. rep., IACR Eprint, 2007. http:/eprint.iacr.org/ or http:/www.seas.gwu.edu/~poorvi/cgrv2007.pdf.

[45] Clark, J., Essex, A., and Adams, C. Secure and observable auditing of electronic voting systems using stock indices. In *Proceedings of the 2007 IEEE Canadian Conference on Electrical and Computer Engineering* (2007).

[46] Clarkson, W., Weyrich, T., Finkelstein, A., Heninger, N., Halderman, J. A., and Felten, E. W. Fingerprinting blank paper using commodity scanners. In *Proceedings of the IEEE Symposium on Security and Privacy* (May 2009).

[47] Coney, L. Report on the Manual Ballot Audit: Takoma Park, Maryland, November 3 2009 Election, 19 November 2009. Electronic Privacy Information Center, http://epic.org/privacy/voting/takoma_park_audit.pdf.

[48] Conrad, F. G., Bederson, B. B., Lewis, B., Peytcheva, E., Traugott, M. W., Hanmer, M. J., Herrnson, P. S., and Niemi, R. G. Electronic voting eliminates hanging chads but introduces new usability challenges. *Int. J. Hum.-Comput. Stud. 67*, 1 (2009), 111–124.

[49] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms, 2nd edition.* MIT Press, McGraw-Hill Book Company, 2000.

[50] CRANOR, L. F., AND GARFINKLE, S. *Security and Usability: Designing Secure Systems that People Can Use.* O'Reilly Media, 2005.

[51] CUNNINGHAM-GLEN, W. *The ballot act, 1872, with copious notes and index.* Shaw and Sons, Fetter Lane, Latin Printers and Publishers, London, UK, 1878.

[52] DEVEGILI, A. J. Farnel: Uma proposta de protocolo criptográco para votação digital, 2001.

[53] E. VINCENT CROSS, I., MCMILLIAN, Y., GUPTA, P., WILLIAMS, P., NOBLES, K., AND GILBERT, J. E. Prime iii: a user centered voting system. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems* (2007), pp. 2351–2356.

[54] ELECTION DATA SERVICES, I. Voting equipment summary by type. http://www.electiondataservices.com/images/File/VotingEquipStudies%20/ve2006%5Creport.pdf, 2006.

[55] ELGAMAL, T. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory 31*, 4 (1985), 469–472.

[56] ESSEX, A. Punchscan: Designing an Independent Verification Mechanism for Elections. Master's thesis, University of Ottawa, Ottawa, Canada, December 2007.

[57] ESSEX, A., CLARK, J., CARBACK, R., AND POPOVENIUC, S. Punchscan in Practice: an E2E Election Case Study. In *Proceedings of the 2007 IAVoSS Workshop on Trustworthy Elections* (2007).

[58] ESSEX, A., CLARK, J., CARBACK, R., AND POPOVENIUC, S. The Punchscan Voting System: VoComp competition submission. In *Online Proceedings of the First University Voting Systems Competition (VoComp)* (2007).

[59] ESTEHGHARI, S., AND DESMEDT, Y. Exploiting the Client Vulnerabilities in Internet E-voting Systems: Hacking Helios 2.0 as an Example. In *EVT/WOTE'10: Proceedings of the Electronic Voting Technology Workshop/Workshop on Trustworthy Elections* (Berkeley, CA, USA, 2010), USENIX Association/IAVoSS/ACCURATE.

[60] EVERETT, S. P. *The Usability of Electronic Voting Machines and How Votes Can Be Changed Without Detection.* PhD thesis, Rice University, May 2007.

[61] FELLER, J., FITZGERALD, B., HISSAM, S. A., AND LAKHAMI, K. R., Eds. *Perspectives on Free and Open Source Software.* MIT Press, 2005.

[62] 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 Information Forensics and Security: Special Issue on Electronic Voting 4* (December 2009), 628–637.

[63] FINK, R. A., SHERMAN, A. T., AND CHALLENER, D. C. A human attestation protocol for trustworthy electronic voting: bootstrapping trust using TPMs, smart cards, timings, and scratch-off codes. Unpublished manuscript, June 2010.

[64] FISHER, K. Punchscan: Security Analysis of a High Integrity Voting System. Master's thesis, University of Maryland, Baltimore County, Baltimore, MD, USA, December 2006.

[65] FISHER, K., CARBACK, R., AND SHERMAN, A. T. 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.

[66] FISHKIN, J. S. *The voice of the people: public opinion and democracy.* Yale University Press, New Haven, CT, USA, 1995.

[67] FOR JUSTICE AT NYU SCHOOL OF LAW, B. C. Recommendations of the Brennan Center for Justice and the Leadership Conference on Civil Rights for Improving Reliability of Direct Recording Electronic Voting Systems. http://www.brennancenter.org/content/resource/recommendations_for_improving_reliability_of_direct_recording_electronic_vo/, June 2004.

[68] FREE SOFTWARE FOUNDATION, INC. GNU General Public License, version 2. http://www.gnu.org/licenses/gpl-2.0.html, June 1991.

[69] FRESOLONE, M. Tactile ballots alternative voting method for the blind. http://www.votersunite.org/info/tactileballots.asp; last accessed 25 September, 2009.

[70] GARDNER, R., YASINSAC, A., BISHOP, M., KOHNO, T., HARTLEY, Z., KERSKI, J., GAINEY, D., WALEGA, R., HOLLANDER, E., AND GERKE, M. Software Review and Security Analysis of the Diebold Voting Machine Software. Tech. rep., Security and Assurance in Information Technology Laboratory, Florida State University, Tallahassee, Florida, July 2007.

[71] GARDNER, R. W., GARERA, S., AND RUBIN, A. D. Detecting code alteration by creating a temporary memory bottleneck. *IEEE Transactions on Security and Forensics 4*, 4 (2009).

[72] GONGGRIJP, R., HENGEVELD, W.-J., HOTTING, E., SCHMIDT, S., AND WEIDEMANN, F. RIES - Rijnland Internet Election System: A Cursory Study of Published Source Code. In *E-Voting and Identity*, P. Ryan and B. Schoenmakers, Eds., vol. 5767 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, pp. 157–171. 10.1007/978-3-642-04135-8_10.

[73] GROUP, T. C. TCG TPM Specification Version 1.2, Revision 103. Available at https://www.trustedcomputinggroup.org/specs/TPM, 2008. Last accessed on Mar 15, 2008.

[74] HERRNSON, P. S., BEDERSON, B. B., HADLEY, C. D., NIEMI, R. G., AND HANMER, M. J. A Study of Vote Verification Technology Conducted for the Maryland State Board of Elections Part II: Usability Study. http://www.elections.state.md.us/voting_system/verification.html, January 2006.

[75] HERRNSON, P. S., NIEMI, R. G., HANMER, M. J., BEDERSON, B. B., CONRAD, F. C., AND TRAUGOTT, M. W. *Voting Technology: The Not-So-Simple Act of Casting a Ballot.* Brookings Institution Press, 2008.

[76] HERRNSON, P. S., NIEMI, R. G., HANMER, M. J., BEDERSON, B. B., CONRAD, F. G., AND TRAUGOTT, M. The Importance of Usability Testing of Voting Systems. In *EVT'06: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop* (Berkeley, CA, USA, 2006), USENIX Association.

[77] HIBBERT, C. *The Days of the French Revolution.* William Morrow, 1999.

[78] HOSP, B., POPOVENIUC, S., SIMHA, R., STANTON, J., AND VORA, P. Implementation and evaluation of a cryptographically secure voting system. Available at http:/vote.cs.gwu.edu/, December 2005.

[79] HOSP, B., AND VORA, P. An Information-Theoretic Model of Voting Systems. In *Preproceedings of the 2006 IAVoSS Workshop on Trustworthy Elections (WOTE 2006)* (Robinson College, Cambridge, United Kingdom, 2006), International Association for Voting System Sciences.

[80] HUBBERS, E., JACOBS, B., AND PIETERS, W. RIES - Internet Voting in Action. *Computer Software and Applications Conference, Annual International 1* (2005), 417–424.

[81] IBM CORPORATION. The Trusted Computing Software Stack (TrouSerS) software library. Available at http://trousers.sourceforge.net, 2008. Last accessed April, 2008.

[82] JAKOBSSON, M., JUELS, A., AND RIVEST, R. Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking. In *Proceedings of the 11th USENIX Security Symposium* (San Francisco, CA, USA, 2002), Usenix Assoc., pp. 339–353.

[83] JONES, D. Voting on Paper Ballots. Available at http://www.cs.uiowa.edu/~jones/voting/paper.html. Last accessed October, 2010.

[84] JONES, D. W. Threats to voting systems. *NIST Workshop on Threats to Voting Systems* (October 2005).

[85] JONES, D. W. On Optical Mark-Sense Scanning. In *Towards Trustworthy Elections: New Directions in Electronic Voting*, D. Chaum, M. Jakobsson, R. L. Rivest, P. Y. A.

Ryan, J. Benaloh, M. Kutylowski, and B. Adida, Eds., vol. 6000 of *Lecture Notes in Computer Science*. IAVOSS/Springer-Verlag Berlin Heidelberg, New York, NY, USA, 2010, pp. 175–190.

[86] KARLOF, C., SASTRY, N., AND WAGNER, D. Cryptographic Voting Protocols: A Systems Perspective. In *Proceedings of the 14th USENIX Security Symposium* (August 2005).

[87] KELLER, A. M., CHERLIN, E., AND MERTZ, D. A Deeper Look: Rebutting Shamos on e-Voting. In *Online Proceedings of the First University Voting Systems Competition (VoComp)* (2007).

[88] KELSEY, J., REGENSCHEID, A., MORAN, T., AND CHAUM., D. Hacking paper: Some random attacks on paper-based E2E systems. In *Frontiers of Electronic Voting* (2007).

[89] KIAYIAS, A., MICHEL, L., RUSSELL, A., AND SHVARTSMAN, A. A. Security Assessment of the Diebold Optical Scan Voting Terminal, Oct. 2006.

[90] KINGDOM, U. The ballot reform act of 1832, 1832.

[91] KOHNO, T., STUBBLEFIELD, A., RUBIN, A. D., AND WALLACH, D. S. Analysis of an Electronic Voting System, May 2004.

[92] 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.

[93] LASKOWSKI, S. Improving the Usability and Accessibility of Voting Systems and Products. Special Publications SP 500-256, NIST, 2004.

[94] MÄGI, T. Practical Security Analysis of E-voting Systems. Master's thesis, Tallinn University, Tallinn, Estonia, 2007.

[95] MERCURI, R. *Electronic Vote Tabulation Checks and Balances*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, October 2000.

[96] MICHELS, AND HORSTER. Some remarks on a receipt-free and universally verifiable mix-type voting scheme. In *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology* (1996), LNCS, Springer-Verlag.

[97] MILLER, J., AND KROSNICK, J. The impact of candidate name order on election outcomes. *Public Opinion Quarterly 62*, 3 (1998), 291.

[98] MORAN, T., AND NAOR, M. Split-ballot voting: everlasting privacy with distributed trust. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security* (2007), pp. 246–255.

[99] MYERS, A. C., CLARKSON, M., AND CHONG, S. Civitas: Toward a secure voting system. In *IEEE Symposium on Security and Privacy* (2008), pp. 354–368.

[100] NECHVATAL, J. *Public-Key Cryptography*. National Computer Systems Lab, Gaithersburg, Maryland, April 1991.

[101] NEFF, C. A. Practical high certainty intent verification for encrypted votes, 2004.

[102] NEFF, C. A. Verifiable mixing (shuffling) of El-Gamal pairs. http:/www.votehere.net/vhti/documentation, April 2004.

[103] NEWKIRK, M. G. Trends in American trust in voting technology. Tech. rep., InfoSentry, 2004.

[104] NEWMAN, T. Tasmania and the Secret Ballot. *Australian Journal of Politics and History 49*, 1 (2003), 93–101.

[105] NOLL, L. C., MENDE, R. G., AND SISODIYA, S. Method for seeding a pseudo-random number generator with a cryptographic hash of a digitization of a chaotic system. US Patent, Mar 1998. 5,732,138.

[106] NORRIS, D. F. Maryland Registered Voters' Opinions About Voting and Voting Technologies. Tech. rep., National Center for the Study of Elections of the Maryland Institute for Policy Analysis & Research University of Maryland, Baltimore County, 2006.

[107] NORRIS, D. F., SEARS, A., NICHOLAS, C., GANGOPADHYAY, A., HOLDEN, S. H., KARABATIS, G., KORU, A. G., LAW, C. M., PINKSTON, J., SHERMAN, A. T., ZHANG, D., DALE, M., FISHER, K., HECKLE, R., KHOO, K., PERLMAN, E., SHEIKH, M. A., ZIMMERMAN, T., AND UNACHUKWU, C. A Study of Vote Verification Technologies Part I: Technical Study. http://www.elections.state.md.us/voting_system/verification.html, February 2006.

[108] OF NEW JERSEY LAW DIVISION-MERCER COUNTY, S. C. Docket No.:MER-L-2691-04 Civil Action Opinion. http://euro.ecom.cmu.edu/people/faculty/mshamos/GuscioraOpinion.pdf, February 2010.

[109] PAILLIER, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *EUROCRYPT* (1999), pp. 223–238.

[110] Paul, N., and Tanenbaum, A. S. Trustworthy voting: From machine to system. *Computer 42*, 5 (2009), 23–29.

[111] Pearson, S., and Balacheff, B. *Trusted computing platforms: TCPA technology in context.* Prentice Hall PTR, 2003.

[112] Pedersen, T. P. A threshold cryptosystem without a trusted party. In *Advances in Cryptology - EuroCrypt '91* (1991), pp. 522–526.

[113] Peisch, P. J. Procurement and the Polls: How Sharing Responsibility for Acquiring Voting Machines Can Improve and Restore Confidence in American Voting Systems. *The Georgetown Law Journal 97*, 3 (March 2009), 878–915.

[114] Peters, R. A. *A Secure Bulletin Board.* PhD thesis, Technische Universiteit Eindhoven, Eindhoven, Netherlands, June 2005.

[115] Popoveniuc, S. *A Framework for Secure Mixnet-Based Electronic Voting.* PhD thesis, The George Washington University, Washington, DC, USA, August 2009.

[116] Popoveniuc, S., and Carback, R. ClearVote: An End-to-End Voting System that Distributes Privacy Between Printers. In *WPES '10: Proceedings of the 9th annual ACM workshop on Privacy in the electronic society* (New York, NY, USA, 2010), ACM, pp. 119–122.

[117] Popoveniuc, S., Clark, J., Carback, R., and Essex, A. Securing Optical-Scan Voting. In *Towards Trustworthy Elections: New Directions in Electronic Voting*, D. Chaum, M. Jakobsson, R. L. Rivest, P. Y. A. Ryan, J. Benaloh, M. Kutylowski, and B. Adida, Eds., vol. 6000 of *Lecture Notes in Computer Science.* IAVOSS/Springer-Verlag Berlin Heidelberg, New York, NY, USA, 2010, pp. 357–369.

[118] POPOVENIUC, S., AND HOSP, B. 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.

[119] POPOVENIUC, S., AND HOSP, B. An introduction to punchscan. In *Proceedings of the 2006 IAVoSS Workshop on Trustworthy Elections* (2006).

[120] 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.

[121] POPOVENIUC, S., AND STANTON, J. Undervote and pattern voting: Vulnerability and a mitigation technique. In *Preproceedings of the 2007 IAVoSS Workshop on Trustworthy Elections (WOTE 2007)* (June 2007).

[122] POUNDSTONE, W. *Gaming the Vote: Why Elections Aren't Fair (and What We Can Do About It).* Hilll and Wang, New York, NY, USA, 2008.

[123] RIVEST, R., AND WACK, J. On the notion of "software independence" in voting systems. DRAFT Version Retrieved on September 25, 2007.

[124] RIVEST, R. L. The ThreeBallot Voting System. http:/theory.csail.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf, October 2006.

[125] 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.

[126] RIVEST, R. L., AND SMITH, W. D. Three voting protocols: threeballot, VAV, and twin. In *Proceedings of the 2007 USENIX/ACCURATE Electronic Voting Technology Workshop* (2007).

[127] ROBERS, H. Electronic elections employing DES smartcards. Master's thesis, Delft University of Technology, Delft, Netherlands, December 1998.

[128] ROBERTSON, J. Security chip that does encryption in pcs hacked. USA Today, Feb 8, 2010. Available at http://usat.ly/9gdlNR. Last accessed October 25, 2010.

[129] RÖSSLER, T., LEITOLD, H., AND POSCH, R. E-voting: A scalable approach using xml and hardware security modules. *e-Technology, e-Commerce, and e-Services, IEEE International Conference on 0* (2005), 480–485.

[130] SAKO, K., AND KILIAN, J. Receipt-free mix-type voting scheme: a practical solution to the implementation of a voting booth. In *Advances in Cryptology - EUROCRYPT'95* (1995).

[131] SALTMAN, R. G. *The History and Politics of Voting Technology: In Quest of Integrity and Public Confidence.* Palgrave Macmillan, January 2006.

[132] 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).

[133] SCIENCE APPLICATIONS INTERNATIONAL CORPORATION, S. Risk Assessment Report Diebold AccuVote-TS Voting System and Process. http:/www.verifiedvoting.org/downloads/votingsystemreportfinal.pdf, September 2003.

[134] SESHADRI, A., PERRIG, A., VAN DOORN, L., AND KHOSLA, P. SWATT: SoftWare-based ATTestation for embedded devices. In *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on* (2004), IEEE, pp. 272–282.

[135] 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.

[136] 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.

[137] SHERMAN, A. T., GANGOPADHYAY, A., HOLDEN, S. H., KARABATIS, G., KORU, A. G., LAW, C. M., NORRIS, D. F., PINKSTON, J., SEARS, A., AND ZHANG, D. An Examination of Vote Verification Technologies: Findings and Experiences from the Maryland Study. In *EVT'06: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop* (April 2006), USENIX Association.

[138] City of Takoma Park, Maryland City Election November 3, 2009 Certification of Election Results, November 2009. http://www.takomaparkmd.gov/clerk/election/2009/results/2009cert.pdf.

[139] TALBERT, R. J. A. *Plutarch on Sparta*. The Penguin Group, London, England, 1988.

[140] United States Election Assistance Commission (EAC). The 2005 Voluntary Voting System Guidelines. http://www.eac.gov/testing_and_certification/2005_vvsg.aspx, December 2005.

[141] University of California, Lawrence Berkeley Laboratory. Berkely systems distribution license. http://www.xfree86.org/3.3.6/COPYRIGHT2.html#5, 1993.

[142] Wang, K., Rescorla, E., Shacham, H., and Belongie, S. OpenScan: A Fully Transparent Optical Scan Voting System. 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.

[143] Winckler, M., Bernhaupt, R., Palanque, P., Lundin, D., Leach, K., Ryan, P. Y. A., Alberdi, E., and Strigini, L. Assessing the usability of open verifiable E-voting systems: a trial with the system Prêt à Voter. In *Proceedings of the International Conference on eGovernment and eGovernance* (Ankara, Turkey, March 2009).

[144] Wolchok, S., Wustrow, E., Halderman, J. A., Prasad, H. K., Kankipati, A., Sakhamuri, S. K., Yagati, V., and Gonggrijp, R. Security analysis of India's electronic voting machines. In *CCS '10: Proceedings of the 17th ACM conference on Computer and communications security* (New York, NY, USA, 2010), ACM, pp. 1–14.