

# JETS

The USENIX Journal of  
Election Technology and Systems

Volume 2, Number 3 • July 2014



**usenix**

THE ADVANCED  
COMPUTING SYSTEMS  
ASSOCIATION

# JETS

## The USENIX Journal of Election Technology and Systems

Volume 2, Number 3 • July 2014

Every Vote Counts: Ensuring Integrity in Large-Scale Electronic Voting .....	1
Feng Hao, <i>Newcastle University</i> ; Matthew N. Kreeger, <i>Thales E-Security</i> ; Brian Randell, Dylan Clarke, Siamak F. Shahandashti, and Peter Hyun-Jeen Lee, <i>Newcastle University</i>	
Usability of Voter Verifiable, End-to-end Voting Systems: Baseline Data for Helios, Prêt à Voter, and Scantegrity II.....	26
Claudia Z. Acemyan, Philip Kortum, Michael D. Byrne, and Dan S. Wallach, <i>Rice University</i>	
Mitigating Coercion, Maximizing Confidence in Postal Elections.....	57
Jacob Quinn Shenker and R. Michael Alvarez, <i>California Institute of Technology</i>	

### JETS Editorial Board

#### Editors-in-Chief

Walter Mebane, *University of Michigan*  
Dan S. Wallach, *Rice University*

#### Editorial Board

Vittorio Addona, *Macalester College*  
Ben Adida, *Mozilla Foundation*  
R. Michael Alvarez, *California Institute of Technology*  
Mary Batcher, *Ernst & Young*  
Josh Benaloh, *Microsoft Research*  
Stephen Checkoway, *Johns Hopkins University*  
Jeremy Clark, *Carleton University*  
Gustavo Delfino, *Universidad Central de Venezuela*  
Jeremy Epstein, *SRI International and National  
Science Foundation*  
Kentaro Fukumoto, *Gakushuin University*  
James Heather, *University of Surrey*  
Michael C. Herron, *Dartmouth College*  
F. Daniel Hidalgo, *Massachusetts Institute of  
Technology*  
Candice Hoke, *Cleveland-Marshall College of Law*  
Joseph Kiniry, *Danmarks Tekniske Universitet*  
Philip Kortum, *Rice University*  
Martha Kropf, *University of North Carolina, Charlotte*  
Sharon Laskowski, *National Institute of Standards  
and Technology*  
Joseph Lorenzo Hall, *Center for Democracy and  
Technology*  
Tal Moran, *Interdisciplinary Center Herzliya*  
Olivier Pereira, *Université catholique de Louvain*  
Maria Petrova, *New Economic School, Moscow*  
Ronald Rivest, *Massachusetts Institute of  
Technology*  
Mark D. Ryan, *University of Birmingham*  
Peter Ryan, *University of Luxembourg*  
Hovav Shacham, *University of California, San Diego*  
Alexander A. Shvartsman, *University of Connecticut*  
Alberto Simpser, *University of Chicago*  
Philip Stark, *University of California, Berkeley*  
Bob Stein, *Rice University*  
Charles Stewart, *Massachusetts Institute of  
Technology*  
Wendy Tam Cho, *University of Illinois, Urbana-  
Champaign*  
Vanessa Teague, *University of Melbourne*  
Alexander Treschel, *European University Institute*  
Melanie Volkamer, *Technische Universität Darmstadt*  
David Wagner, *University of California, Berkeley*  
Douglas Wikström, *KTH Royal Institute of  
Technology*

JETS articles will be presented at the Electronic Voting Technology  
Workshop/Workshop on Trustworthy Elections (EVT/WOTE).

[www.usenix.org/conferences/evtwote](http://www.usenix.org/conferences/evtwote)

©2014 by The USENIX Association

All Rights Reserved

This volume is published as a collective work. Rights to individual papers remain with the author or the author's employer. Permission is granted for the noncommercial reproduction of the complete work for educational or research purposes. USENIX acknowledges all trademarks herein.

ISBN 978-1-931971-14-0 ISSN 2328-2797



## Every Vote Counts: Ensuring Integrity in Large-Scale Electronic Voting

Feng Hao, Newcastle University, UK  
Matthew N. Kreeger, Thales E-Security, UK  
Brian Randell, Newcastle University, UK  
Dylan Clarke, Newcastle University, UK  
Siamak F. Shahandashti, Newcastle University, UK  
Peter Hyun-Jeen Lee, Newcastle University, UK

This paper presents a new End-to-End (E2E) verifiable e-voting protocol for large-scale elections, called Direct Recording Electronic with Integrity (DRE-i). In contrast to all other E2E verifiable voting schemes, ours does not involve any Tallying Authorities (TAs). The design of DRE-i is based on the hypothesis that existing E2E voting protocols' universal dependence on TAs is a key obstacle to their practical deployment. In DRE-i, the need for TAs is removed by applying novel encryption techniques such that after the election multiplying the ciphertexts together will cancel out random factors and permit anyone to verify the tally. We describe how to apply the DRE-i protocol to enforce the tallying integrity of a DRE-based election held at a set of supervised polling stations. Each DRE machine directly records votes just as the existing practice in the real-world DRE deployment. But unlike the ordinary DRE machines, in DRE-i the machine must publish additional audit data to allow public verification of the tally. If the machine attempts to cheat by altering either votes or audit data, then the public verification of the tallying integrity will fail. To improve system reliability, we further present a fail-safe mechanism to allow graceful recovery from the effect of missing or corrupted ballots in a publicly verifiable and privacy-preserving manner. Finally, we compare DRE-i with previous related voting schemes and show several improvements in security, efficiency and usability. This highlights the promising potential of a new category of voting systems that are E2E verifiable and TA-free. We call this new category "self-enforcing electronic voting".

### 1. INTRODUCTION

**Background.** An electronic voting (e-voting) system is a voting system in which the election data is recorded, stored and processed primarily as digital information [VoteHere 2002]. Depending on the implementation, e-voting can be either local or remote. Local e-voting occurs at a supervised polling station, normally using a touch-screen machine to record votes directly. Such a machine is often called a Direct Recording Electronic (or DRE) machine [Kohno et al. 2004]. In contrast, remote e-voting can be conducted at any location, usually through a web browser [Adida 2008; Adida et al. 2009].

E-voting has already been widely deployed across the world. As shown in USA Today [Wolf 2008], the use of DRE expanded rapidly in the United States following the 2000 national election: from 12% of the votes cast in that election to 29% in 2004, and to 38% in 2006. India moved to full DRE voting in their 2004 national election, and Brazil started its first fully DRE-based election in 2002 [Blanc 2007]. In 2007, Estonia became the first country to allow Internet voting for national elections [Krimmer et al. 2007]. Many other countries have been actively pursuing the implementation of e-voting [Alvarez et al. 2011; Pieters 2011].

**Controversy.** However, e-voting has become controversial. In 2004, Kohno *et al.* critically analysed a type of e-voting machine that had been widely used in the US, and discovered serious software vulnerabilities and bugs [Kohno et al. 2004]. The alarming level of security flaws was especially worrying because the U.S. government had earlier certified the machine to be "trustworthy". In response to these and other similar findings [Sherman et al. 2006; Jefferson et al. 2004] regarding other manufacturers' machines, many people have demanded that e-voting be abandoned completely. Several U.S. states consequently abandoned the use of e-voting machines in 2008, causing a rapid decline of DRE usage from 38% in 2006 to 32% in 2008 [Wolf 2008]. Similar problems have also been reported in other countries, e.g., Germany, Netherlands and Ireland have all sus-

---

This work is supported by the European Research Council (ERC) Starting Grant (No. 106591) on "Self-enforcing electronic voting: trustworthy elections in the presence of corrupt authorities".

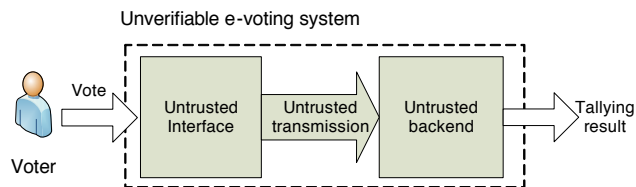


Fig. 1. An unverifiable (black-box) e-voting system

pending e-voting in 2008-2009 [Alvarez et al. 2011; Pieters 2011]. In 2010, researchers also started to seriously question the integrity of e-voting machines used for elections in India [Wolchok et al. 2010].

A fundamental problem with many deployed e-voting systems (including the discarded/suspended ones) is that they are *unverifiable* [Pieters 2011]. Essentially each system works like a black-box (Figure 1). After voting, the voter has no means of telling whether her vote was correctly recorded. At the end of the election, the system announces the tallied result for each candidate, but any independent verification of this result is impossible.

Typically, a black-box e-voting system comprises three components: a voting interface, a transmission mechanism and a tallying back-end (see Figure 1). The voting interface may be a touch screen in a local DRE-based election, or it may be a web browser in a remote Internet-based election. In either case, a compromised voting interface (touch-screen DRE or a web browser) may surreptitiously change the voter’s choice; the transmission of electronic votes (either off-line or on-line) may be intercepted and the votes modified; and the back-end may maliciously change the tally to support some particular candidate regardless of the actual vote count. In summary, there are many opportunities for an attacker to tamper with the electronic data without the public being aware of the change.

This can be contrasted with elections that involve votes being recorded on a visible physical medium such as a printed paper ballot form or a punched card. The processing of such votes can be easily and effectively monitored (e.g., by multiple independent poll-watchers, both professionals and amateurs). And these votes can be retained in case of a challenge, and if necessary be recounted. However, similar direct physical monitoring is not possible in electronic voting.

Government certification of an e-voting system’s hardware and software was perceived by many countries as the solution to the problem of achieving trustworthy e-voting [Alvarez et al. 2011; Pieters 2011], but has proved inadequate for several reasons. First of all, it requires people to trust the probity and competence of the certification authority. Second, it does not solve the fundamental problem, because a certified black-box is still a black-box (i.e., its operation is unverifiable). Third, researchers have repeatedly demonstrated that attackers can successfully compromise “certified” e-voting systems, altering election results without their activities being detected [Sherman et al. 2006; Kohno et al. 2004]. All these greatly reduce public confidence in such government certification.

Therefore, for e-voting to succeed in the future, it is important that the voting system be verifiable [Adida 2008; Adida et al. 2009]. However, it is worth noting that the idea of verifiable e-voting is not new; it has existed for over twenty years [Benaloh 1987]. Although progress has been made in trialling verifiable e-voting protocols in practice [Chaum et al. 2008a; Adida et al. 2009], so far the impact on real-world national elections has been limited. In practice, many countries around the world are still using unverifiable (black-box) e-voting systems.

**E2E verifiability.** To explain the limitations of existing verifiable e-voting technology, we first need to clarify what is meant by being “verifiable”. In general verifiability has two levels of meaning: individual and universal [Chaum et al. 2008a]. At the individual level, all voters should be able to verify that their votes have been correctly recorded and have been correctly included into the tally. At the universal level, anyone in the world should be able to verify the integrity of the tallying result, based on publicly available audit data. E-voting systems that satisfy the verifiability at both levels

are generally termed as being End-to-End (E2E) verifiable [Adida 2008]. We refer the reader to papers by Küsters *et al.* [Küsters and Vogt 2010] and Popoveniuc *et al.* [Popoveniuc *et al.* 2010] for more formal definitions of E2E verifiability.

Some researchers suggested adding a Voter Verifiable Paper Audit Trail (VVPAT) to a DRE machine. Most notably, the method proposed by Mercuri [Mercuri 2001] works as follows: when the voter makes a selection on the touch-screen, the machine prints the selected choice on a paper receipt in plain text. The voter can visually inspect the receipt under a layer of glass before it is automatically transferred to a secure location. The voter is not allowed to take the receipt home as that would reveal (to a coercer) how she had voted. Overall, this method improves the individual verifiability by allowing voters to check if their votes have been recorded correctly. Also, it provides a physical paper trail that permits a manual recount in case of a dispute. However, the VVPAT method provides no means for voters to check whether the recorded votes will be securely transported to the tallying unit and whether the votes will be tallied correctly. Therefore, a DRE system based on VVPAT alone is not E2E verifiable.

Thus, the dual, and potentially conflicting, challenges faced by the designers of any voting system are to ensure the system is publicly verifiable and meanwhile to preserve the voter's privacy. To satisfy the E2E verifiability, it is necessary to provide the voter a receipt, which can be checked against a public bulletin board [Chaum *et al.* 2008a]. In order to prevent coercion and vote selling, the receipt must not reveal any information about how the voter had voted. On the other hand, if the receipt does not show how the voter had voted, how can she be sure it is a correct record of her vote? These requirements may seem clearly contradictory, but past research has shown that they can be met by combining various techniques, e.g., cryptography and voter-initiated auditing [Adida *et al.* 2009; Benaloh 2007].

To date, many E2E verifiable voting protocols have been proposed. Well-known examples include: Adder [Kiayias *et al.* 2006], Civitas [Clarkson *et al.* 2008], Helios [Adida 2008; Adida *et al.* 2009], Scantegrity [Chaum *et al.* 2008b], Scantegrity II [Chaum *et al.* 2008a], Prêt à Voter [Ryan *et al.* 2009], MarkPledge [Adida and Neff 2006] and Chaum's visual cryptographic scheme [Chaum 2004]. All these protocols rely on there being multiple independent Tallying Authorities (TAs) to perform and control the tallying process in a publicly verifiable manner. Hence, we choose to categorise them as "TA-based E2E verifiable e-voting".

Protocols in this category generally work as follows (Figure 2): the voter, using a voting interface, casts a vote and obtains a receipt. The receipt is encrypted under a set of tallying authorities' public keys (or one joint public key). At the end of the election, the system publishes all the receipts on a public bulletin board (e.g., a mirrored public web site), so that voters can check if their votes have been recorded. However, individual voters are unable to decrypt their receipts to confirm their votes have been correctly recorded. Instead they are provided with some other (indirect) way of gaining confidence that this is the case (through voter-initiated auditing, as we will explain in Section 2).

Since all the data on the bulletin board is encrypted, the tallying authorities are needed to perform the decryption and tallying process. This process can be done in a publicly verifiable manner, so that the TAs do not need to be trusted for the integrity of the tallying result. However, they need to be trusted to some extent for the secrecy of individual votes. The common mitigating measure is to put the TAs under a  $k/n$  threshold control, where  $n$  is the total number of TAs and  $k$  is the threshold. Only if more than a threshold  $k$  number of TAs are corrupted will they be able to decrypt each individual vote. Furthermore, it is normally assumed that the TAs are selected from different parties with conflicting interests, hence they supposedly lack the incentive to collude. (Nonetheless, it is important to ensure the TAs use *independent* software, because "if all trustees (TAs) use tallying software from a single source, then this software might collude without the trustees' knowledge." [Karlof *et al.* 2005])

**Implementing E2E verifiability.** Although many TA-based E2E verifiable voting protocols have been proposed, only a few have actually been implemented in practice. The Helios voting system is notable for being the first web-based implementation of an E2E verifiable voting system. Initially, Helios (v1.0) used mix-net based tallying [Adida 2008], and later it (v2.0) was changed to using

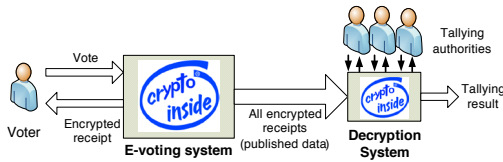


Fig. 2. TA-based e-voting with E2E verifiability

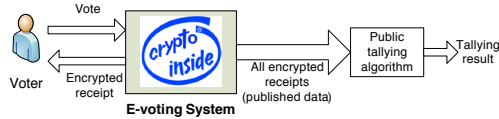


Fig. 3. Self-enforcing e-voting with E2E verifiability

homomorphic tallying [Adida et al. 2009]. In 2009, a customized variant of Helios 2.0 was adopted by the Université catholique de Louvain (UCL) in a campus election to elect the university president.

As highlighted in the Helios paper [Adida et al. 2009], the practical implementation of tallying authorities has proved to be “a particularly difficult issue”. To ensure the fairness in the representations, the tallying authorities were chosen from various groups (students, administrative staff and so on) with different backgrounds (not just computer science). However, it turned out that the chosen authorities did not have the required technical expertise to perform complex cryptographic operations. Hence, a group of “external experts” (whose identities are not mentioned in the Helios paper [Adida et al. 2009]) were invited to first perform the key generation on behalf of the tallying authorities. The whole procedure included purchasing brand new laptops, removing the hard disk drives, disabling wireless network cards, booting up the machines using standard linux live-CDs and loading the key generation code (written in Python) through the USB sticks. Subsequently, the tallying authorities’ private keys were generated and stored on the USB sticks, which were then distributed to the authorities. In the mean time, all of the generated private keys were centrally backed up by one trusted third party (a notary public). After the election, “a similar procedure was followed when those keys were used for decryption” [Adida et al. 2009]. Clearly, the tallying authorities’ further dependence on “external experts” and a single trusted third party for backup has significantly complicated the trust relationships in the election management.

**Removing TAs.** A few researchers have investigated how to remove tallying authorities in electronic voting. Kiayias and Yung first studied this in 2002 with a boardroom voting protocol [Kiayias and Yung 2002], followed by Groth in 2004 [Groth 2004] and Hao-Ryan-Zieliński in 2010 [Hao et al. 2010]. Among these boardroom voting protocols, the Hao-Ryan-Zieliński’s solution [Hao et al. 2010] is so far the most efficient in every aspect: the number of rounds, the computation load and the message size. In general, a boardroom voting protocol works by requiring voters to cooperatively interact with all other voters in a network in a number of rounds. In the best case [Hao et al. 2010], only two rounds of interactions are needed. The tallying result is usually computed by voters through exhaustive search. Essentially, the voting is totally decentralized and run by the voters themselves. A decentralized boardroom voting protocol, such as Kiayias-Yung’s [Kiayias and Yung 2002], Groth’s [Groth 2004], or Hao-Ryan-Zieliński’s [Hao et al. 2010], can provide the theoretically-best protection of ballot secrecy. In order to learn a voter’s secret choice, the attacker must compromise all other voters to form a full collusion against the voter [Kiayias and Yung 2002; Groth 2004; Hao et al. 2010].

A boardroom voting protocol is considered different from an E2E verifiable voting protocol for a number of reasons. First of all, they differ on the scales. The former is usually designed for small-scale voting in a boardroom, while the latter is normally for large-scale country voting. Using exhaustive search to determine the tally may be straightforward in boardroom voting, but it may prove expensive if the election is a large-scale one (especially for multi-candidate elections). Second, the system infrastructures are different. The former is decentralized; voters use their own *trusted* computing hardware/software to interact with all other voters through a fully connected network. There is no voter-receipt (as there is no entity to issue receipts) and there is no central bulletin board to check receipts [Hao et al. 2010]. The latter is centralized; there is little interaction between voters. People vote through some common voting interface (e.g., touch-screen DRE). A voter normally gets a receipt, which can be compared against a central bulletin board. Third, the security

requirements are completely different. For example, in a boardroom voting protocol [Kiayias and Yung 2002; Groth 2004; Hao et al. 2010], a voter can trivially prove to a coercer how she had voted by revealing the ephemeral secret generated during the protocol. Furthermore, any arbitrary voter can easily disrupt a multi-round voting procedure by simply dropping out half-way in the protocol. While coercion, vote selling and voter disruption might not be considered serious issues in a small boardroom, they are important considerations in the design of an E2E verifiable voting system.

The scope of this paper is to focus on E2E verifiable voting systems for large-scale elections. Existing boardroom voting protocols are clearly unsuitable for any country-scale elections. However, they are still relevant to our study as they demonstrate that it is possible to remove TAs albeit only in the setting of a small-scale election. To the best of our knowledge, no one has investigated the feasibility of removing tallying authorities for large-scale elections. Indeed, existing E2E verifiable e-voting protocols designed for large-scale elections universally require involving external tallying authorities in the tallying process [Kiayias et al. 2006; Clarkson et al. 2008; Adida 2008; Adida et al. 2009; Chaum et al. 2008b; Chaum et al. 2008a; Ryan et al. 2009; Adida and Neff 2006; Chaum 2004].

**Contributions.** We initiate a study on whether it is feasible to remove the dependence on external tallying authorities in an E2E verifiable voting system. Along this direction, we propose to replace the tallying authorities and the decryption system in Figure 2 by a public algorithm. We define the resultant system as a “self-enforcing e-voting” system (see Figure 3). Because the algorithm is public, the tallying process is fully verifiable without any TA involvement. The main contributions of this paper are summarized below:

- We present the first E2E verifiable voting protocol that is TA-free. Our protocol is called Direct Recording Electronic with Integrity (DRE-i). Its “self-enforcing” property is realized by integrating a cancellation formula [Hao and Zieliński 2006] into the homomorphic tallying process: the encryption of votes follows a well-defined structure such that after the election multiplying the ciphertexts together will cancel out random factors and permit anyone to verify the tally. A similar tallying method was used in a previous Hao-Ryan-Zieliński boardroom voting protocol [Hao et al. 2010], but ours does not require exhaustive search. Although the two protocols share the same mathematical formula for cancelling random factors, they are designed for completely different election scenarios and have different security requirements.
- We effectively combine the basic DRE-i with several additional engineering designs to make it an overall secure and practical system, suitable for a DRE-based election at polling stations. The first is to seamlessly integrate the voter’s initiated auditing into the natural confirm/cancel voting experience on a touch-screen DRE. As a result, the system is user-friendly to a voter who does not understand cryptography at all. Furthermore, we provide a fail-safe mechanism to allow graceful recovery of partially corrupted audit data in a publicly verifiable and privacy-preserving way. Finally, we support a distributed computation of secret keys to distribute trust and improve system availability. (Advantages of our system over previous ones will be detailed in Section 4.)

## 2. A SELF-ENFORCING E-VOTING PROTOCOL

In this section, we describe a self-enforcing e-voting protocol called Direct Recording Electronic with Integrity (DRE-i). In particular, we show how to apply the DRE-i protocol to enforce the tallying integrity of DRE-based local voting at the polling station. (It is possible to implement DRE-i for remote voting [Hao et al. 2012; Hao et al. 2013], but to avoid confusion, we will focus on local voting in this paper.) For the simplicity of discussion, we will consider a single-candidate election first, and then extend it to multiple candidates.

### 2.1. User roles

In an E2E verifiable e-voting protocol, there are generally three user roles as defined below [Adida et al. 2009].

- (1) **Ordinary voter:** Someone who directly participates in the voting.

- (2) **Auditor:** Someone who audits the system by performing real-time checks on the system during the voting process.
- (3) **Universal verifier:** Anyone in the world who has the technical expertise to verify the audit data published by the voting system.

## 2.2. Integrity requirements

We also adopt the commonly accepted integrity requirements for an E2E verifiable voting protocol [Adida et al. 2009; Chaum et al. 2008b; Chaum et al. 2008a].

- (1) **Ballot format integrity:** Everyone, including third party observers, should be able to verify that every encrypted ballot has the correct format to represent exactly one vote.
- (2) **Ballot casting integrity:** All voters should be able to convince themselves that their cast ballots are recorded to the correct candidates.
- (3) **Ballot transmission integrity:** All voters should be able to verify that their recorded ballots have been correctly transmitted to the tallying process.
- (4) **Ballot tallying integrity:** Everyone, include third party observers, should be able to verify that the tallying result is correctly obtained from the recorded ballots.

Obviously, the integrity requirements must be satisfied without compromising the voter's privacy. In particular, the receipt that permits a voter to verify the integrity of the voting system must not reveal how she had voted. We will explain in Section 3 that this holds true in DRE-i.

## 2.3. Trust assumptions

There are many other requirements to make a secure e-voting system. Since the satisfaction of those requirements is generally assumed in the literature [Kiayias et al. 2006; Clarkson et al. 2008; Adida 2008; Adida et al. 2009; Chaum et al. 2008b; Chaum et al. 2008a; Adida and Neff 2006; Chaum 2004], we make the same assumptions, namely:

- (1) **User enrolment:** Only eligible users can be enrolled in the voter registration.
- (2) **User authentication:** Only authenticated voters are allowed to vote during the election.
- (3) **One-man-one-vote:** Each authenticated voter is allowed to vote just once.
- (4) **Voting privacy:** Voting happens in a private space that no one else can observe.
- (5) **Anonymity:** The voting machine that is used does not know the real identity of the voter.
- (6) **Public bulletin board:** There is a publicly readable, append-only bulletin board (e.g., a mirrored public website), on which the legitimate voting system can publish audit data for verification (the authenticity of data can be ensured by the use of digital signatures).

If voting takes place in a supervised environment (say a polling station), it is relatively easy to meet the above assumptions. For example, the polling station staff can authenticate voters based on their ID documents or even biometrics. After successful authentication, the voter is free to take a single random authentication token, say a smart card. The voter then enters a private booth and uses the token to authenticate herself to the machine and starts voting [Kohn et al. 2004]. To ensure one-person-one-vote, the polling station can publish a list of the names of the people who voted, so that anyone can verify that the number of voters matches the number of cast votes [Chaum et al. 2008a]. Observers at a polling station can also independently count how many people have actually voted.

## 2.4. Three Stages of Voting

The DRE-i protocol consists of three phases: setup, voting and tallying. The following sections explain each phase in detail.

*2.4.1. Setup phase.* We describe the protocol in a multiplicative cyclic group setting (i.e., DSA-like group), though the same protocol also works in an additive cyclic group (i.e., ECDSA-like group). Let  $p$  and  $q$  be two large primes, where  $q | p - 1$ .  $\mathbb{Z}_p^*$  is a multiplicative cyclic group and



Table I. Setup phase before election

Ballot No	Random public key	Restructured public key	Cryptogram of no-vote	Cryptogram of yes-vote
1	$g^{x_1}$	$g^{y_1}$	$g^{x_1 \cdot y_1}$ , 1-of-2 ZKP	$g^{x_1 \cdot y_1} \cdot g$ , 1-of-2 ZKP
2	$g^{x_2}$	$g^{y_2}$	$g^{x_2 \cdot y_2}$ , 1-of-2 ZKP	$g^{x_2 \cdot y_2} \cdot g$ , 1-of-2 ZKP
...	...	...	...	...
$n$	$g^{x_n}$	$g^{y_n}$	$g^{x_n \cdot y_n}$ , 1-of-2 ZKP	$g^{x_n \cdot y_n} \cdot g$ , 1-of-2 ZKP

Note: Data in the first three columns are published on a public bulletin board before the election. They serve as commitment so that the values cannot be later changed. Data in the last two columns are kept secret; they are either computed on-demand during voting or pre-computed before the election.

$G_q$  its subgroup of prime order  $q$ . Let  $g$  be the generator of  $G_q$  (any non-identity element in  $G_q$  can serve as a generator). We assume the Decision Diffie-Hellman (DDH) problem [Stinson 2006] in  $G_q$  is intractable. The parameters  $(p, q, g)$  are publicly agreed before the election starts. Unless the contrary is stated explicitly, all the modular operations are performed with respect to the modulus  $p$ . Hence, we omit the explicit “mod  $p$ ” for simplicity.

First of all, the DRE machine generates a private signing key, say using DSA or ECDSA [Stinson 2006], and publishes the public key on the bulletin board. A tamper-resistant module is used to securely manage the private signing key, in line with industry standard practice [Anderson 2008]. The private signing key is generated on-board in the secure memory of the module and never leaves the protected boundary of the module.

Subsequently, the DRE machine computes a table as shown in Table I. The table contains  $n$  rows with each row corresponding to a ballot, so there are  $n$  ballots in total. The number  $n$  is the product of the total number of the eligible voters and a safety factor ( $> 1$ ). The safety factor, say 10, is defined so as to allow the generation of extra ballots for auditing purposes (as we will explain later).

Each row in Table I corresponds to a ballot with encrypted data (cryptograms) to represent candidate choices. In a single-candidate election, the choices are “Yes” and “No”. All rows are constructed to satisfy four properties. First, given any cryptogram in any row, one can easily verify that it is an encryption of one of the two values: “Yes” or “No” (which translate to 1 and 0 in the implementation). Second, given only a single cryptogram from any selected row, one cannot tell whether it is “Yes” or “No”. Third, given both cryptograms (unordered) from any selected row, anyone will be able to easily tell which is “Yes” and which is “No”. Fourth, given a set of cryptograms, each of which was arbitrarily selected, one from each row, one can easily check how many “Yes” values in total are in the set. In the following, we will explain how these four properties are fulfilled and how they are useful in building a self-enforcing e-voting system.

The system fills the table as follows. For each of the  $n$  ballots, the system computes a random public key  $g^{x_i}$ , where  $x_i \in_R [1, q - 1]$ . When this has been done for all the ballots, the system computes  $g^{y_i} = \prod_{j < i} g^{x_j} / \prod_{j > i} g^{x_j}$  for every ballot. Here, we call the obtained  $g^{y_i}$  a restructured public key, because it is constructed by multiplying all the random public keys before  $i$  and dividing the result by all the public keys after  $i$ . Note that anyone is able to compute  $g^{y_i}$  based on the published  $g^{x_i}$  values.

The “Yes”/“No” value in each ballot is encoded in the form of  $C_i = g^{x_i y_i} \cdot g^{v_i}$  where  $v_i = 0$  for “No” and 1 for “Yes”. The no-vote,  $g^{x_i y_i}$ , is indistinguishable from random based on the DDH assumption (detailed proofs can be found in Section 3). Clearly, the yes-vote,  $g^{x_i y_i} \cdot g$ , is indistinguishable from random too. However, if both no-vote and yes-vote are published, then it is trivial to distinguish which is “No” and which is “Yes” (because the latter is the former multiplied by  $g$ ).

In addition, the system needs to compute a 1-out-of-2 Zero Knowledge Proof (ZKP) for each yes/no value. This is to ensure that the value of the vote is indeed in the correct form of  $C_i = g^{x_i y_i} \cdot g^{v_i}$  where  $v_i \in \{0, 1\}$ . In other words, the value  $v_i$  can only be one of: 0 and 1. We adopt the standard 1-out-of- $n$  ZKP technique (also known as the CDS protocol) due to Cramer, Damgård and Schoenmakers [Cramer et al. 1994]. Although the original CDS protocol is designed for ElGamal encryption, it is directly applicable here if we regard  $g^{y_i}$  as a public key. (The only difference is

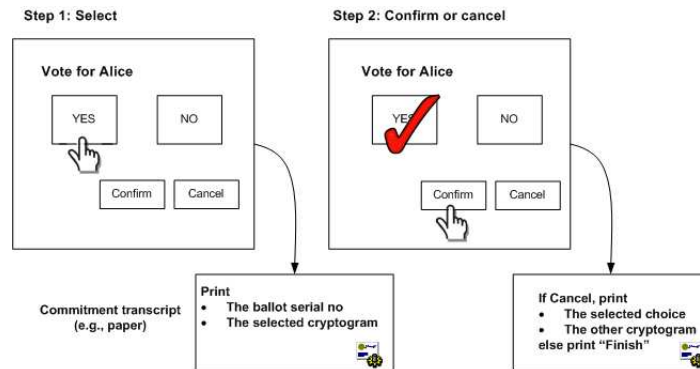


Fig. 4. A simple single-candidate voting interface. The receipt has two parts: the first includes the printout in Step 1 with a digital signature and the second includes the printout in Step 2 with a signature that covers the entire transcript.

that the public key in ElGamal encryption is statically fixed, while in our case, it is dynamically constructed from  $g^{x_i}$  values for each ballot.) Here, we use  $n = 2$ . The original three-move interactive CDS protocol can be made non-interactive by applying the standard Fiat-Shamir heuristics [Fiat and Shamir 1987]. The same CDS technique has been widely used in previous e-voting protocols to ensure the published ciphertext is well-formed.

As shown in Table I, the cryptogram of the no-vote contains  $g^{x_i y_i}$  and a 1-out-of-2 ZKP; similarly, the cryptogram of the yes-vote comprises  $g^{x_i y_i} \cdot g$  and a corresponding 1-out-of-2 ZKP.

Similar to the private signing key, all  $x_i$  secrets are generated on-board in the module and are stored within the module's secure memory. The corresponding public keys ( $g^{x_i}$ ) are published on the bulletin board before the election; they serve as commitment so the values cannot be changed later. To ensure authenticity, all commitment data published on the bulletin board should be digitally signed. Let us assume  $n = 10^5$  and a group setting of 2048-bit  $p$  and 256-bit  $q$ . The total size of  $x_i$  secrets is 3.2 MB. Hence, it is possible to store the  $x_i$  secrets entirely in the module's memory. (As an example, a high capacity smart card can have 16 MB non-volatile memory.)

In order to optimize the performance in voting, one may choose to pre-compute all the cryptograms (last two columns in Table I) before the election. In that case, the secrecy of pre-computed cryptograms needs to be protected at the same level as the  $x_i$  secrets. If the size of the cryptograms is more than what the module's memory can accommodate, one solution, as commonly adopted in industry, is to generate a master key on-board in the module and use the master key to encrypt blobs of data in an authentic manner, so that the encrypted blobs can be stored outside the module and be reloaded back to memory when needed [Anderson 2008]. This is a typical trade-off between memory and speed. Reloading the blob to memory will involve some decryption work, but since it is only a symmetric-key operation, it can be very fast.

**2.4.2. Voting phase.** As stated before, we assume the eligible voter has been properly authenticated. She first obtains a random authentication token, enters a private voting booth, uses the token to authenticate herself to the machine and starts voting. The voter is prompted to select a choice on a touch screen DRE (see Figure 4). To cast her ballot, the voter follows two basic steps below.

In step one, the voter selects a choice on the screen. Meanwhile, the machine prints the following data on the paper: the ballot serial number  $i$ , and the cryptogram of the selected choice. (The ballot serial number  $i$  may be incremental or randomly assigned; there is no significant difference from the protocol's perspective as long as the number is unique.) The printed data serve as a commitment, as it cannot be changed. The commitment transcript is digitally signed by the machine to prove its authenticity. As explained earlier, the machine's public key is publicly announced before the election, so the signature is universally verifiable.

Table II. Ballot tallying.

No <i>i</i>	Random pub key $g^{x_i}$	Restructured pub key $g^{y_i}$	Published Votes $V_i$	ZKPs
1	$g^{x_1}$	$g^{y_1}$	Valid: $g^{x_1 \cdot y_1}$	a 1-of-2 ZKP
2	$g^{x_2}$	$g^{y_2}$	Valid: $g^{x_2 \cdot y_2} \cdot g$	a 1-of-2 ZKP
3	$g^{x_3}$	$g^{y_3}$	Dummy: $g^{x_3 \cdot y_3}, g^{x_3 \cdot y_3} \cdot g$	two 1-of-2 ZKPs
...	...	...	...	...
<i>n</i>	$g^{x_n}$	$g^{y_n}$	Dummy: $g^{x_n \cdot y_n}, g^{x_n \cdot y_n} \cdot g$	two 1-of-2 ZKPs

*Note:* This entire table is published on the public bulletin board. A vote can be either valid or dummy. Ballot No. 1 shows an example of a valid “No” vote, and No. 2 shows an example of a valid “Yes” vote. Tallying involves multiplying all the  $V_i$  values (only including the “No” votes for the dummy case).

In step two, the voter has the option of either confirming or cancelling the previous selection. If she chooses to confirm, the system will print a “finish” message on the paper, and a valid encrypted vote has been cast. On the other hand, if she chooses to cancel, the DRE machine will reveal the selected choice in plain text (“Yes” or “No”), and also print the other cryptogram on the paper. In this case, a dummy vote has been cast. The touch screen will return to the previous step and provide another unused ballot. Voters are entitled to cast as many dummy votes as they wish<sup>1</sup>, but are allowed to cast only a single valid vote.

The confirm/cancel option in step two serves to provide ballot casting assurance, namely: the voter needs to gain confidence that her actual vote has been recorded as she intended. For example, a corrupted machine might cheat by swapping the “No”/“Yes” cryptograms. The solution here is to have the machine initially commit to a value, and then give the voter an option to challenge the machine to reveal the commitment so that if the machine has cheated, it will be caught once the voter chooses to audit. Successful cheating on any large scale without being detected is extremely unlikely. Our auditing procedure is consistent, in spirit, with Benaloh’s idea of voter-initiated challenges [Benaloh 2007], but it has been more tightly integrated into the overall cryptographic system starting with the initial setup.

The commitment transcript, signed by the machine, for the entire voting session can be printed on a single piece of paper, which forms the voter’s receipt. The data on the receipt is also available on the public bulletin board. The voter is free to take home the receipt and compare it against the bulletin board, so gaining a degree of trust in the bulletin board’s contents. (This is just as in other verifiable e-voting protocols [Kiayias et al. 2006; Clarkson et al. 2008; Adida 2008; Adida et al. 2009; Chaum et al. 2008b; Chaum et al. 2008a; Adida and Neff 2006; Chaum 2004]). When all the voters have cast their votes, or the election time limit is up, the system will publish both the yes-vote and no-vote cryptograms for the remaining unused ballots and mark them as “dummy” votes.

**2.4.3. Tallying phase.** Tallying the ballots involves multiplying together all the published cryptograms  $V_i$  (for dummy votes, using only the *no*-vote; see Table II). Thus, we have:

$$\prod_i V_i = \prod_i g^{x_i y_i} g^{v_i} = \prod_i g^{v_i} = g^{\sum_i v_i}$$

The key to the tallying process is the fact that  $\sum_i x_i y_i = 0$  (a cancellation formula first introduced in 2006 in the design of an anonymous veto protocol [Hao and Zieliński 2006]; we refer the reader to that paper for the proof). Thus, all random factors cancel each other out. Here, we combine this cancellation technique with the conventional homomorphic encryption to build a self-enforcing e-voting protocol. Compared with the existing mix-net or homomorphic aggregation based tallying methods, the new method has the distinctive feature of not requiring any secret keys (hence no TAs).

The term  $\sum_i v_i$  is the total number of the “yes” votes. Note that we do not need to compute the exponent of  $g^{\sum_i v_i}$  (although this is doable by exhaustive search). Because the DRE system records the ballots directly, it announces the count of “yes” votes,  $\beta$ , right after the election, as is current

<sup>1</sup>In practice, a reasonable upper limit would be enforced.

practice in DRE-based elections. Anyone can verify whether  $g^\beta$  and  $g^{\sum_i v_i}$  are equal. This takes only one exponentiation. Also, anyone can count the number of dummy votes from the bulletin board, which we denote as  $\lambda$ . Thus, the tally of “no” votes is  $\alpha = n - \beta - \lambda$ .

There are several ways to extend a single-candidate election to multiple candidates. One straightforward method is to have a Yes/No selection for each of the candidates [Hao et al. 2010]. Another method involves defining more efficient encoding values for candidates [Cramer et al. 1996]. These are standard techniques to extend a single-candidate election to a multiple-candidate election, while the underlying voting protocol remains unchanged.

### 3. SYSTEM ANALYSIS

In this Section, we analyze the DRE-i protocol with regard to security, efficiency, usability and dependability.

#### 3.1. Security analysis

First of all, we show the encryption of the “No” vote is semantically secure: in other words, the value  $g^{x_i y_i}$  for the  $i$ th ballot is indistinguishable from random. As explained earlier, the system selects random values  $x_i \in_R [1, q - 1]$  for  $i = 1, \dots, n$ . The value  $y_i$  is defined from:  $g^{y_i} = \prod_{j < i} g^{x_j} / \prod_{j > i} g^{x_j}$ , hence  $y_i = \sum_{j < i} x_j - \sum_{j > i} x_j$ . Given that  $x_i$  is random,  $y_i \neq 0$  holds with an overwhelming probability (i.e.,  $1 - 1/q$ ). Furthermore,  $y_i$  is random over  $[1, q - 1]$  and it is independent of  $x_i$ , the value  $g^{x_i y_i}$  will be uniformly distributed over non-identity elements in  $G$  [Stinson 2006]. Therefore, the term  $g^{x_i y_i}$  is indistinguishable from random based on the DDH assumption as long as the  $x_i$  values are kept secret. All the  $g^{x_i y_i}$  values ( $i \in [1, n]$ ) are related by the constraint that  $\prod_i g^{x_i y_i} = 1$ . In the following, we will prove that such a structural relationship does not reveal any information other than the tally.

**ASSUMPTION 1 (DDH VARIANT).** For a generator  $g$  and  $a, b \in_R [1, q - 1]$ , given a tuple  $(g, g^a, g^b, C)$  in which  $C$  is either  $g^{ab}$  or  $g^{ab+1}$ , it is hard to decide whether  $C = g^{ab}$  or  $C = g^{ab+1}$ .

**LEMMA 3.1.** Assumption 1 is implied by the DDH assumption (i.e., the problem is at least as hard as the DDH problem).

**PROOF.** Consider the following tuples:

$$(g, g^a, g^b, g^{ab}), \quad (g, g^a, g^b, R), \quad (g, g^a, g^b, R'g), \quad \text{and} \quad (g, g^a, g^b, g^{ab}g),$$

for random  $a, b, R$ , and  $R'$ . DDH guarantees that the first and second tuples are indistinguishable. The second and third tuples have the exact same distribution and hence are indistinguishable. DDH also guarantees that the third and fourth tuples are indistinguishable. Hence, the first and fourth tuples, i.e.  $(g, g^a, g^b, g^{ab})$  and  $(g, g^a, g^b, g^{ab+1})$  are indistinguishable.  $\square$

**Definition 3.2 (Bare Bulletin Board).** A bare bulletin board is a bulletin board without the ZKPs and digital signatures.

In the following analysis, we will first consider a bare bulletin board for simplicity, assuming the underlying ZKPs and digital signature schemes are secure primitives. The ZKPs serve to prove that the ciphertexts published on the bulletin board are well-formed, and they do not reveal any information about the plaintext votes. The digital signatures serve to prove that all data published on the bulletin board are authentic; they are not related to the secrecy of votes.

**LEMMA 3.3.** Consider two DRE-i elections in which all the votes are exactly the same except for two votes  $v_i$  and  $v_j$  which are swapped between the two elections. Under Assumption 1, the bare bulletin boards of these two elections are indistinguishable to an adversary that has the capability to determine an arbitrary number of votes other than  $v_i$  and  $v_j$ .

**PROOF.** If the adversary is one of the voters, he is able to define his own vote. To make it general, we assume a more powerful adversary who can define an arbitrary number of votes except two:  $v_i$

Table III. The simulated bare bulletin boards in the proof of Lemma 3.3.

<b>k</b>	$g^{x_k}$	$g^{y_k}$	$g^{x_k y_k} \cdot g^{y_k}$
1	$g^{v_1}$	$1/\prod_{k>1} g^{x_k}$	$g^{x_1 y_1} \cdot g^{v_1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i$	$g^a$	$\prod_{k<i} g^{x_k} / \prod_{k>i} g^{x_k}$	$(g^a)^{\sigma_i} \cdot g/C$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$j$	$g^b$	$\prod_{k<j} g^{x_k} / \prod_{k>j} g^{x_k}$	$(g^b)^{\sigma_j} \cdot C$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n$	$g^{x_n}$	$\prod_{k<n} g^{x_k}$	$g^{x_n y_n} \cdot g^{y_n}$

 $\Leftrightarrow$ 

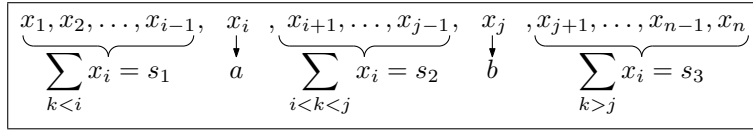
<b>k</b>	$g^{x_k}$	$g^{y_k}$	$g^{x_k y_k} \cdot g^{y_k}$
1	$g^{v_1}$	$1/\prod_{k>1} g^{x_k}$	$g^{x_1 y_1} \cdot g^{v_1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i$	$g^a$	$\prod_{k<i} g^{x_k} / \prod_{k>i} g^{x_k}$	$(g^a)^{\sigma_i} \cdot g/C$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$j$	$g^b$	$\prod_{k<j} g^{x_k} / \prod_{k>j} g^{x_k}$	$(g^b)^{\sigma_j} \cdot C$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n$	$g^{x_n}$	$\prod_{k<n} g^{x_k}$	$g^{x_n y_n} \cdot g^{y_n}$

Note: The two tables are identical except that  $C = g^{ab}$  in one table and  $C = g^{ab+1}$  in the other. They are indistinguishable as long as the two  $C$  values are indistinguishable.

and  $v_j$ . Let us assume w.l.o.g. that  $i < j$ . If  $v_i = v_j$ , the lemma holds trivially. In the following we give a proof for  $v_i \neq v_j$ .

Let us assume there is an adversary  $\mathcal{A}$  that first chooses an arbitrary number of the votes other than  $v_i$  and  $v_j$ , and eventually distinguishes the two elections. Given a tuple  $(g, g^a, g^b, C)$ , where  $a, b \in_R [1, q-1]$  and  $C$  equals either  $g^{ab}$  or  $g^{ab+1}$ , we now construct an algorithm  $\mathcal{S}$  that uses  $\mathcal{A}$  to break Assumption 1. The algorithm  $\mathcal{S}$  sets up the bulletin board with the generator  $g$  as below. Let  $I = \{1, \dots, n\} \setminus \{i, j\}$ .

First,  $\mathcal{S}$  chooses  $n-2$  random values  $x_k$  for all  $k \in I$ .  $\mathcal{S}$  sets  $g^{x_i} \leftarrow g^a$ ,  $g^{x_j} \leftarrow g^b$ , and calculates  $g^{x_k}$  for all  $k \in I$ . Note that we implicitly have  $x_i = a$  and  $x_j = b$ . Let  $s_1 = \sum_{k<i} x_k$ ,  $s_2 = \sum_{i<k<j} x_k$ , and  $s_3 = \sum_{k>j} x_k$ .  $\mathcal{S}$  also calculates  $s_1$ ,  $s_2$ , and  $s_3$  and then computes  $\sigma_i = s_1 - s_2 - s_3$  and  $\sigma_j = s_1 + s_2 - s_3$ . Figure 5 illustrates the relations between  $x_k$  values and  $a$ ,  $b$ ,  $s_1$ ,  $s_2$ , and  $s_3$ .


 Fig. 5.  $x_i$  values used in the simulation

Now given all  $g^{x_k}$ , all  $g^{y_k}$  can be computed accordingly. Note that we implicitly have:

$$y_i = \sum_{k<i} x_k - \sum_{k>i} x_k = s_1 - (s_2 + b + s_3) = \sigma_i - b$$

$$y_j = \sum_{k<j} x_k - \sum_{k>j} x_k = (s_1 + a + s_2) - s_3 = \sigma_j + a$$

$\mathcal{A}$  chooses a set of votes  $\{v_k\}_{k \in I_{\mathcal{A}}}$  for the set of indexes  $I_{\mathcal{A}} \subseteq I$ . Let us consider some arbitrary set of votes  $\{v_k\}_{k \in I \setminus I_{\mathcal{A}}}$ .  $\mathcal{S}$  can calculate  $g^{x_k y_k}$  for all  $k \in I$ , since it knows  $x_k$  and  $g^{y_k}$ . Hence, it can calculate  $g^{x_k y_k} g^{y_k}$  for all  $k \in I$ . For  $k = i, j$ ,  $\mathcal{S}$  sets

$$g^{x_i y_i} g^{y_i} \leftarrow (g^a)^{\sigma_i} \cdot g/C \quad \text{and} \quad g^{x_j y_j} g^{y_j} \leftarrow (g^b)^{\sigma_j} \cdot C.$$

Now the calculation of the entire bare bulletin board is complete. Table III shows the simulated bare bulletin board.

In the case that  $C = g^{ab}$ , we have:

$$g^{x_i y_i} g^{v_i} \leftarrow (g^a)^{\sigma_i} \cdot g / C = (g^a)^{\sigma_i} \cdot g / g^{ab} = g^{a(\sigma_i - b)} g = g^{x_i y_i} g \quad \text{and}$$

$$g^{x_j y_j} g^{v_j} \leftarrow (g^b)^{\sigma_j} \cdot C = (g^b)^{\sigma_j} \cdot g^{ab} = g^{b(\sigma_j + a)} = g^{x_j y_j} ,$$

which means that in our bare bulletin board  $v_i = 1$  and  $v_j = 0$ .

In the case that  $C = g^{ab+1}$ , we have:

$$g^{x_i y_i} g^{v_i} \leftarrow (g^a)^{\sigma_i} \cdot g / C = (g^a)^{\sigma_i} \cdot g / g^{ab+1} = g^{a(\sigma_i - b)} = g^{x_i y_i} \quad \text{and}$$

$$g^{x_j y_j} g^{v_j} \leftarrow (g^b)^{\sigma_j} \cdot C = (g^b)^{\sigma_j} \cdot g^{ab+1} = g^{b(\sigma_j + a)} g = g^{x_j y_j} g ,$$

which means that in our bare bulletin board  $v_i = 0$  and  $v_j = 1$ .

$\mathcal{S}$  then gives  $\mathcal{A}$  the constructed bare bulletin board as input. If  $\mathcal{A}$  is able to distinguish which of the above two cases the given bare bulletin board corresponds to,  $\mathcal{S}$  will be able to successfully distinguish the two cases for  $C$  and hence break Assumption 1.  $\square$

**THEOREM 3.4 (MAIN THEOREM).** *We term the votes that are determined by the adversary “the adversarial votes” and the rest “the non-adversarial votes”. Under the DDH assumption and that the ZKP primitive used in the protocol is secure, the DRE- $i$  bulletin board does not reveal anything about the secrecy of the votes other than the tally of non-adversarial votes to an adversary that is able to determine an arbitrary number of votes.*

**PROOF.** We first restrict our attention to the bare bulletin board and consider the additional ZKP and digital signatures later. To prove that the bare bulletin board does not reveal anything other than the tally of non-adversarial votes, we prove that given only a tally of non-adversarial votes  $t_H$  and a set of adversarial votes  $\{v_k\}_{k \in I_{\mathcal{A}}}$ , a bare bulletin board can be simulated which is indistinguishable from any other bare bulletin board with the same non-adversarial vote tally and given adversarial votes. We do this in two steps: first, we show how to simulate a random bare bulletin board with the same  $t_H$  and given adversarial votes; and second, we show that such a random bare bulletin board is indeed indistinguishable from any other bare bulletin board with the same non-adversarial vote tally and given adversarial votes.

*Step 1.* Given the adversarial votes  $\{v_k\}_{k \in I_{\mathcal{A}}}$ , we randomly choose the rest of the votes  $\{v_k\}_{k \notin I_{\mathcal{A}}}$  such that their tally is  $t_H$ . Choosing random values for  $x_k$  for all  $k$ , we can simulate a bare bulletin board with  $\{v_k\}_{k \in I_{\mathcal{A}}}$  as the adversarial votes and  $\{v_k\}_{k \notin I_{\mathcal{A}}}$  as the non-adversarial votes.

*Step 2.* Consider any two possible bare bulletin boards  $BB$  and  $BB'$  with the same non-adversarial vote tally and given adversarial votes as above. First note that  $BB$  and  $BB'$  have the same adversarial votes, they have the same adversarial vote tally  $t_{\mathcal{A}}$ , and since they have the same non-adversarial vote tally  $t_H$  as well, they have the same total tally  $t = t_H + t_{\mathcal{A}}$ . We know that any two bulletin boards with the same total tally (and hence  $BB$  and  $BB'$ ) differ on an *even* number of votes. Let this vote difference between  $BB$  and  $BB'$  be  $2d$ . This means that with  $d$  swaps, one can get from one bare bulletin board to the other. Lemma 3.3 guarantees that in all these  $d$  steps, under Assumption 1, the two bare bulletin boards involved are indistinguishable to an adversary choosing  $\{v_k\}_{k \in I_{\mathcal{A}}}$ . Note that the adversarial votes remain fixed between the swaps. Furthermore, Assumption 1 is implied by DDH according to Lemma 3.1. Hence, a standard hybrid argument implies that the original bare bulletin boards ( $BB$  and  $BB'$ ) are indistinguishable under the DDH assumption.

A secure 1-of-2 ZKP [Cramer et al. 1994], by definition, does not reveal any information more than the one-bit truth of the statement: whether the ciphertext is a correct encryption of one of the two values. Hence, it does not reveal the secrecy of the encrypted value. The digital signatures serve

to prove that all data published on the bulletin board are authentic; they are not related to the secrecy of votes. Hence, we conclude that the theorem holds for the full bulletin board.  $\square$

The Theorem 3.4 guarantees the highest possible privacy level for DRE-i. To see this, note that the election tally is public in any election, and hence an adversary controlling a number of adversarial votes inevitably finds out the non-adversarial vote tally. The above theorem ensures that this inevitable knowledge is the only knowledge the adversary gains and in this sense proves the highest privacy level for DRE-i.

A corollary of the above theorem can be stated as below for a passive adversary that does not determine any votes, but only observes the bulletin board.

**COROLLARY 3.5 (PRIVACY AGAINST PASSIVE ADVERSARIES).** *Under the DDH assumption and that the ZKP primitive used in the protocol is secure, the DRE-i bulletin board does not reveal anything about the secrecy of the votes other than the tally of the votes to a passive adversary.*

Although we have proven that encrypted votes in DRE-i are protected at the highest possible level, it is important to note that breaking encryption is *not* the only way to compromise ballot secrecy. There are other potentially more effective attacks, and security is determined by the weakest link in the chain. For example, an untrustworthy voting interface is one weak link in the chain; a corrupted interface can trivially disclose the voter's secret choice [Karlof et al. 2005; Estehghari and Desmedt 2010]. The setup phase is another potentially weak link. Existing E2E verifiable voting Protocols [Kiayias et al. 2006; Clarkson et al. 2008; Adida 2008; Adida et al. 2009; Chaum et al. 2008b; Chaum et al. 2008a; Adida and Neff 2006; Chaum 2004] generally require a secure setup phase, in which TAs securely generate and distribute key shares. If the setup phase is compromised by attackers, then the *secrecy* of the vote will be breached. These issues also apply to DRE-i. On the other hand, in DRE-i even if the setup phase is completely corrupted, the tallying *integrity* will remain unaffected. This property is claimed for existing E2E verifiable protocols [Chaum et al. 2008a; Adida et al. 2009]. We now explain how this also holds for DRE-i.

We will show DRE-i satisfies all the four integrity requirements as defined in Section 2.2, even if the setup phase is compromised. The use of the CDS technique (i.e., the 1-out-of- $n$  Zero Knowledge Proof) ensures the correct format of the ballot [Cramer et al. 1994], and fulfills the first requirement. The second requirement is satisfied by the voter-initiated auditing (i.e., voter challenge), which is adopted in most verifiable e-voting protocols. The third requirement, that on transmission integrity, is satisfied by the voter being able to check the receipt against the public bulletin board. The fourth requirement, that on tallying integrity, is fulfilled by using homomorphic aggregation combined with the random-factor cancelation, so that anyone is able to verify the tally based on the audit data published on the public bulletin board without relying on any TA. In summary, if an insider attacker attempts to compromise the integrity of the election at any stage, this will most likely be caught by the public because the protocol is E2E verifiable [Adida et al. 2009; Chaum et al. 2008a].

Finally, it is important to ensure that a receipt does not reveal the voter's choice to a coercer. This is a property formally defined as "receipt-freeness" [Delaune et al. 2006]. Previous E2E verifiable voting protocols [Kiayias et al. 2006; Adida 2008; Adida et al. 2009; Chaum et al. 2008b; Chaum et al. 2008a; Adida and Neff 2006; Chaum 2004] generally satisfy this requirement. We explain our protocol conforms to it too. As explained earlier, if the voter chooses to confirm her vote, the receipt does not leak any information about the choice made. If, on the other hand, the voter opts to cancel her vote, the receipt will reveal the selected choice, but the vote will be declared to be a dummy. A dummy vote is of course useless to a would-be coercer.

### 3.2. Performance evaluation

In DRE-i, we pre-compute all random factors used for encryption before the election with the commitment published on the bulletin board. This pre-computation strategy, combined with the cancellation technique, is one key to realizing the "self-enforcing" property of the voting system. The

same strategy also permits pre-computation of all cryptograms, hence optimizing the performance during voting.

We evaluate the system performance by starting from ballot generation. As shown in Table I, we need to compute  $g^{y_i}$  for each ballot. At first glance, this is very expensive, taking approximately  $n$  multiplications to compute  $g^{y_1}$  (recall that  $n$  is the total number of ballots, which may be hundreds of thousands). However, note that  $g^{y_2} = g^{y_1} \cdot g^{x_2} \cdot g^{x_1}$ . More generally,  $g^{y_i} = g^{y_{i-1}} \cdot g^{x_i} \cdot g^{x_{i-1}}$  for  $i > 1$ . Thus, computing  $g^{y_i}$ , for  $i = 2, 3, \dots, n$ , incurs negligible cost.

For each ballot  $i$ , exponentiation is the predominant cost factor. It takes one exponentiation to compute  $g^{x_i}$ , one to compute  $g^{x_i y_i}$  and four<sup>2</sup> to compute the 1-out-2 ZKP [Cramer et al. 1996] for each no/yes vote, totalling ten exponentiations.

In the ballot casting stage, the computational cost incurred by the DRE machine is small. If we opt for the option of pre-computing all cryptograms before the election, the delay imposed of voting would be almost negligible, since the machine merely needs to print out the pre-computed cryptogram according to the voter's choice and sign it with the digital signature key. Obviously, pre-computing the cryptograms would mean we need to do more preparation work for an election, but that seems a worthwhile trade-off.

The data published on the bulletin board is universally verifiable. Anyone is able to check that the published random public keys  $g^{x_i}$  lie within the prime-order group, and that the values of  $g^{y_i}$  are correctly computed. To verify the ZKP for the published vote  $V_i$ , it is necessary to first validate the order of  $V_i$ . This requires an exponentiation (for both the valid and dummy cases); it takes a further four exponentiations to verify the 1-out-of-2 ZKP [Cramer et al. 1994; Cramer et al. 1996]. In total, it takes roughly 5 exponentiations to verify a ZKP. In principle, it suffices for at least one person to verify all the ZKPs in a batch (those who lost the election would be motivated to verify the tally).

### 3.3. Usability

As explained earlier in Section 2.1, there are three types of users in an e-voting system: ordinary voters, auditors and universal verifiers. In the DRE-i protocol, the auditing is voter-initiated, so an ordinary voter is also an auditor. Of course this does not preclude employing dedicated auditors in an election to perform auditing by casting dummy votes. A universal verifier is anyone in the world who has the technical expertise to verify all data on the public bulletin board in a batch operation.

For an e-voting system to be practically useful, it needs to be “usable”. However the notion of “usability” can be abstract and elusive. Here, we define a “usable” cryptographic e-voting system as one that can be used independently by ordinary voters and auditors without requiring any cryptographic knowledge or relying on any trusted software. This is because in practice most people have no knowledge of cryptography and cannot distinguish trustworthy software from untrustworthy software.

The DRE-i protocol assumes a minimum technical background about the voter who may wish to audit the system. The auditing process has been seamlessly integrated into the natural confirm/cancel selection. Every voter can easily audit the ballot by simply choosing the “cancel” button. If a ballot is canceled, the voter just needs to verify that the printed candidate choice (in plain text) on the receipt is the same as that she chose previously. If not, she should lodge a protest immediately. This can be done without requiring any cryptographic knowledge. Of course, the voter needs to know how to open a web browser and check the bulletin board. This basic computer skill is also assumed in other verifiable e-voting protocols [Kiayias et al. 2006; Clarkson et al. 2008; Adida 2008; Adida et al. 2009; Chaum et al. 2008b; Chaum et al. 2008a; Adida and Neff 2006; Chaum 2004].

One may be concerned about the authenticity of the receipt and how to verify this. The data on the receipt should be authentic; otherwise, a dishonest voter may modify the receipt to support a protest that the data fail to match that on the bulletin board. Obviously, if we wish to assume the official receipt paper is physically unforgeable and any tampering with the printed data on the receipt will be visibly evident, then such an attack will not work. However, the assumption of the physical

<sup>2</sup>This is estimated based on using a simultaneous computation technique [Menezes et al. 1996].



unforgeability is difficult to realize. In most cases, a digital signature would be needed, as in other e-voting protocols. With DRE-i, the voter does not have to verify the signature cryptographically; all she needs to do is to ensure the data on the receipt matches that on the bulletin board. A universal verifier will be able to verify all data on the bulletin board in a batch. We assume there is a facility provided at the polling station, say before the exit of the station, to allow voters to check the bulletin board. If the data is found not to match, the voter should raise the matter immediately.

### 3.4. Dependability and fault tolerance

In DRE-i, the integrity of the election tally depends on the accuracy and completeness of the audit data. The DRE machine directly records votes just as the existing practice in real-world DRE deployment. At the end of the election, the machine reports the tally that it counts internally. But unlike the ordinary DRE machines, in DRE-i, the machine must publish additional audit data to allow public verification of the tally. If the audit data is corrupted (say some ballots are lost), then the integrity of the tally will be lost and the universal verification will fail. In that case, the system essentially degenerates to the existing unverifiable DRE-based e-voting. Here, we have considered the assurance of tallying integrity in the most stringent case, ensuring that every vote must be counted.

In a practical election, it is desirable to handle system faults gracefully. When the audit data have been found to be partially corrupted, instead of merely degenerating to unverifiable e-voting, we can extend the DRE-i protocol to provide a fail-safe feature.

**Fail-safe DRE-i.** Consider a case where a small subset  $L$  of ballots are found missing from (or to be corrupted on) the public bulletin board. (The number of the missing ballots should be insufficient to change the election outcome; otherwise, the act of error recovery may not be meaningful.) We assume the DRE machine still maintains the  $x_i$  secrets in the protected memory of the tamper-resistant module. To allow the tallying verification to proceed, one trivial solution is to re-publish the cryptograms of the subset  $L$  of ballots as if they were “dummy” votes. The no-votes ( $g^{x_i y_i}$  for  $i \in L$ ) are then included into the tallying process, hence allowing the tally of the remaining ballots to be verified. However, if a voter holds a receipt of a missing ballot, the secrecy of that ballot will be lost. Hence, instead of publishing individual cryptograms, it is more secure to publish just one aggregate value: namely,  $A = \sum_{i \in L} g^{x_i y_i}$  together with some cryptographic proofs to show that  $A$  is in the correct format (details can be found in Appendix A). Thus, the information leakage is minimal. An attacker in possession of some (not *all*) receipts cannot learn anything about the missing ballots. In the worst case when the attacker is able to collect *all* receipts of the missing ballots, the only thing he can learn is the tally of the missing ballots, not any individual vote.

**Distributed DRE-i.** The fail-safe mechanism works on the condition that the  $x_i$  secrets are available. If the DRE machine is physically damaged or lost, such an error recovery procedure may no longer be possible. In order to ensure system robustness, it is desirable to implement DRE-i in a distributed way, as we explain below.

Figure 6 shows one possible implementation of the DRE-i system using a distributed client-server architecture. The system consists of touch-screen DRE clients and a back-end server cluster. The DRE client interacts with the voter and records the vote directly as usual. The server cluster consists of  $n$  servers and implements a  $k/n$  threshold control. The setup phase works based on a proactive secret sharing scheme [Herzberg et al. 1995]. Each server generates a random polynomial of degree  $t - 1$  and distributes  $n$  shares to all servers. All  $n$  polynomials are then added up with no single server knowing the aggregate secret. Let the aggregate secret be  $x_i$ . The process can be repeated for all  $x_i$  where  $i = 1, \dots, n$ . Subsequently, the server cluster jointly compute  $g^{x_i}$  by performing secret reconstruction on the exponent [Herzberg et al. 1995], such that no single server learns the exponent  $x_i$ . To finish the setup phase, the server cluster publishes all the  $g^{x_i}$  values on the bulletin board as commitment. During the voting phrase, the DRE client queries the shares from  $k$  honest servers in the server cluster through secure channels and reconstructs the  $x_i$  secret. With  $x_i$ , the client is able to compute the cryptogram and print the receipt accordingly. The DRE client erases the transient  $x_i$  secret immediately after its use.

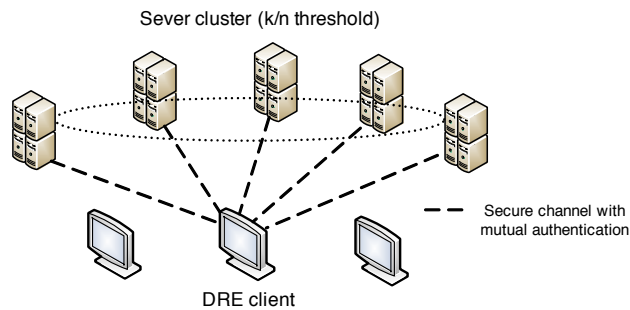


Fig. 6. A distributed implementation of the DRE-i system

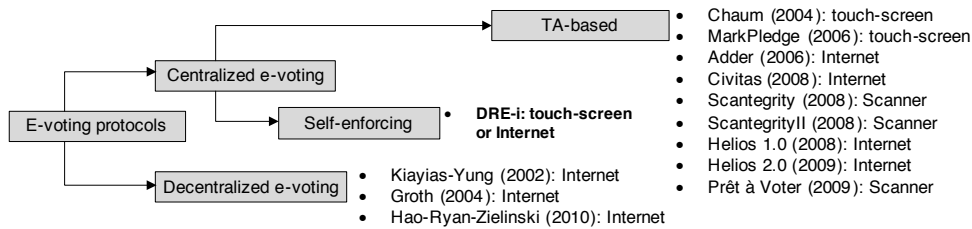


Fig. 7. Categorization of e-voting protocols

A further practical strategy in distributing the implementation of DRE-i is to divide the national-scale tallying into a set of smaller-scale tallying processes, each implementing an independent DRE-i system. This is consistent with many real-world elections where tallies are calculated at relatively small (say county or precinct) scales and then added up.

#### 4. RELATED WORK AND COMPARISON

In this section, we compare DRE-i with previous DRE-based voting protocols in a local supervised voting environment.

##### 4.1. Categorization of e-voting

First of all, we take a broad view at the existing e-voting protocols. There are generally two categories of cryptographic voting protocols: decentralized and centralized (see Figure 7). The former includes boardroom voting protocols due to Kiayias-Yung [Kiayias and Yung 2002], Groth [Groth 2004] and Hao-Ryan-Zieliński [Hao et al. 2010]. The latter includes a wide range of E2E verifiable protocols: e.g., Adder [Kiayias et al. 2006], Civitas [Clarkson et al. 2008], Helios [Adida 2008; Adida et al. 2009], Scantegrity [Chaum et al. 2008b], Scantegrity II [Chaum et al. 2008a], Prêt à Voter [Ryan et al. 2009], MarkPledge [Adida and Neff 2006] and Chaum’s visual cryptographic scheme [Chaum 2004]. Existing E2E verifiable voting protocols are often designed to use different voting interfaces: e.g., a web browser [Kiayias et al. 2006; Adida 2008; Adida et al. 2009], an optical scanner [Chaum et al. 2008b; Chaum et al. 2008a; Ryan et al. 2009], and a touch-screen DRE [Adida and Neff 2006; Chaum 2004]. They are also designed to suit two different scenarios: local voting [Chaum et al. 2008b; Chaum et al. 2008a; Ryan et al. 2009; Adida and Neff 2006; Chaum 2004] and remote voting [Kiayias et al. 2006; Clarkson et al. 2008; Adida 2008; Adida et al. 2009]. All these E2E verifiable protocols require external tallying authorities to decrypt and tally the submitted votes. Hence, they belong to the category of “TA-based e-voting” (see Figure 7).

The proposed DRE-i protocol provides the same E2E verifiability, but *without* involving any external tallying authorities. This puts DRE-i in a new category, which we call “self-enforcing e-voting”.

Table IV. Comparison between DRE-i and ordinary (black-box) e-voting in local DRE-based voting.

	<b>DRE-i</b>	<b>Ordinary (black-box) DRE machine</b>
External tallying authorities	Not required	Not required
Ballot casting assurance	Voter-initiated auditing	<b>No assurance</b>
Transmission integrity	Check receipt with Public Bulletin Board	<b>No assurance</b>
Tallying integrity	Accurate audit data	<b>No assurance</b>
Ballot secrecy	Voting interface, <b>setup</b> and <b>DRE not leaking random factors (or pre-computed cryptograms)</b>	Voting interface
Voter privacy	Anonymity	Anonymity
Receipt	Yes, but cannot be used for coercion	<b>No receipt</b>
Availability	Dependent on system robustness	Dependent on system robustness
Tamper-resistant module	<b>Needed for key management</b>	Not required
Crypto-awareness of voter	Not required	Not required
Crypto-awareness of auditor	Not required	<b>Public auditing is impossible</b>
Crypto-awareness of verifier	Required	<b>Universal verification is impossible</b>

Notes: Major differences are highlighted in bold face.

#### 4.2. Comparison with unverifiable DRE

We first compare DRE-i with the unverifiable (or black-box) DRE machines that have been widely deployed around the world. The results of the comparison are summarized in Table IV. We explain the main differences below.

**Integrity.** The primary advantage of DRE-i lies in the additional “-i” in the name: i.e., its integrity. In DRE-i, a voter can verify that her ballot is recorded to the correct candidate through voter-initiated auditing (i.e., *ballot casting integrity*). She can further verify that the recorded ballot is correctly transmitted to the tallying unit by checking the receipt against the public bulletin board (i.e., *transmission integrity*). Finally, every voter is able to verify the integrity of the tally based on the public audit data published on the bulletin board (i.e., *ballot tallying integrity*). These essential verification procedures are missing in the currently deployed DRE machines.

**Ballot secrecy and voter privacy.** In both systems, the touch-screen interface can violate the secrecy of the vote. However, it does not know the voter’s real identity. Hence, the voter’s privacy is protected through anonymity. In DRE-i, the system requires an additional setup phase, which prefixes random factors used for encryption. The secrecy of the random factors needs to be securely protected, as well as the pre-computed cryptograms (if the pre-computation option is enabled).

**Receipt.** In DRE-i, the machine prints out a receipt, which the voter can verify against a public bulletin board. The receipt does not reveal how a voter had voted, but allows the voter to check if her vote has indeed been included into the tallying process. By contrast, the ordinary DRE machine does not provide any receipt. If the ballot is missing or miscounted, the voter would not be able to know.

**Tamper-resistant module.** In DRE-i, a tamper-resistant module (e.g., smart card or TPM chip) is needed to securely manage sensitive key material, including the private signing key, the  $x_i$  secrets and pre-computed cryptograms (if any). This follows the standard industry practice for key management [Anderson 2008]. However, an ordinary DRE machine normally does not require a tamper-resistant module, as no cryptography is used.

**Usability.** As compared to the ordinary DRE, the usability in DRE-i degrades slightly due to the additional opportunity provided to the voter to check the receipt against the bulletin board. On the other hand, the receipts allow public verification of the tallying integrity, which is not possible with ordinary DRE machines. Hence, the trade-off seems worthwhile for the improved assurance on integrity.

#### 4.3. Comparison with previous DRE-based E2E verifiable schemes

Next, we compare DRE-i with two previous DRE-based E2E verifiable voting protocols: Mark-Pledge [Adida and Neff 2006] and Chaum’s visual crypto scheme [Chaum 2004]. The results of this comparison are summarized in Table V.

Table V. Comparison between DRE-i and related E2E verifiable voting protocols for local DRE-based voting

	<b>DRE-i</b>	<b>Local DRE-based protocols</b> [Adida and Neff 2006; Chaum 2004]
External tallying authorities	Not required	<b>Required</b>
Ballot casting assurance	Voter-initiated auditing	Voter-initiated auditing
Transmission integrity	Check receipt with Public Bulletin Board	Check receipt with Public Bulletin Board
Tallying integrity	Accurate audit data	Accurate audit data, and <b>TA not losing keys</b>
Ballot secrecy	Voting interface, setup, DRE not leaking random factors (or <b>pre-computed cryptograms</b> )	Voting interface, setup, DRE not leaking random factors and <b>TA not leaking private keys</b>
Voter privacy	Anonymity	Anonymity
Receipt-freeness	Yes	Yes
Availability	Dependent on system robustness	Dependent on system robustness and <b>TA not losing keys</b>
Tamper-resistant module	Needed for key management	Needed for key management
Crypto-awareness of voter	Not required	<b>Required</b>
Crypto-awareness of auditor	Not required	<b>Required</b>
Crypto-awareness of verifier	Required	Required

Note: Major differences are highlighted in bold face.

**Integrity.** DRE-i provides the same E2E verifiability as MarkPledge [Adida and Neff 2006] and Chaum’s scheme [Chaum 2004], but without involving any external tallying authorities. To guarantee the tallying integrity, all three protocols require the audit data as published on the bulletin board be accurate and complete. In MarkPledge [Adida and Neff 2006] and Chaum’s scheme [Chaum 2004], when the election is finished, the audit data published on the bulletin board must be first decrypted by external tallying authorities before any verification is possible. This requires that the tallying authorities’ private keys be available at the decryption and tallying phase; otherwise, the tally cannot be verified.

**Ballot secrecy and voter privacy.** In all three protocols, if the voting interface is corrupted, the secrecy of the ballot is lost. In addition, if the setup process (be it the pre-computation procedure in DRE-i or the secret sharing setup in TA-based e-voting) is compromised, the secrecy of the ballot is lost too. In DRE-i, all random factors are pre-determined before the election with commitment published on the bulletin board. The secrecy of the pre-determined random factors needs to be securely protected, which can be realized by storing them in the secure memory of a tamper-resistant module [Anderson 2008]. In MarkPledge [Adida and Neff 2006] and Chaum’s scheme [Chaum 2004], the random factors are generated by the DRE machine on the fly during the encryption of ballots. Similarly, the secrecy of those random factors needs to be protected. It is critically important that the random factors are generated honestly from a secure random number generator. If the random number generator is corrupted, all random factors are effectively leaked. Consequently, the secrecy of all encrypted votes is trivially lost (which is orthogonal to the security of the TAs’ private keys). In DRE-i, the choice of pre-computing random factors before the election is based on the assumption that the environment in the setup phase is more controllable than that in the field deployment on the election day, hence the random number generator is less likely to be corrupted. Finally, MarkPledge [Adida and Neff 2006] and Chaum’s scheme [Chaum 2004] assume the external TAs do not leak their private keys; otherwise, the secrecy of the votes is compromised.

**Availability.** All three protocols depend on the robustness of hardware and software to ensure availability of functionality. In MarkPledge [Adida and Neff 2006] and Chaum’s scheme [Chaum 2004], the tallying process is entirely reliant on the external tallying authorities. All data is encrypted under the authorities’ keys and there is usually no mechanism of directly recording votes by the machine. However, this dependence on external authorities may lead to an additional, in fact a catastrophic, failure mode. Human nature being what it is, when a security system critically depends on a few selected human beings as authorities, they may form the weakest link in the system [Anderson 2008]. Suppose that when the national voting is finished, tallying authorities claim that their private keys are lost [Karlof et al. 2005] (e.g., as victims of targeted attacks or as the au-

thorities claim such is the case). All the data on the bulletin board will be useless, and the whole election may have to be aborted as a result. (Recall that in the Helios election [Adida et al. 2009], all the tallying authorities' private keys were centrally backed up at a trusted third party to ensure availability.)

**Tamper-resistant module.** In DRE-i, a tamper-resistant module is required to securely manage sensitive key material, including the private signing key, the pre-determined random factors and the pre-computed cryptograms (if any). In MarkPledge [Adida and Neff 2006] and Chaum's scheme [Chaum 2004], a tamper-resistant module is also required, for safeguarding the private signing key, and additionally, for protecting the ephemeral random factors that are generated as part of the encryption process. Because of the pre-generation of random factors, DRE-i requires more memory in the tamper-resistant module than the other two schemes.

**Usability.** In MarkPledge, the voter needs to supply a "short-string challenge" [Adida and Neff 2006], which demands special cryptographic knowledge. To address this limitation, the designers of the MarkPledge system suggest having a trusted third party at the polling station to issue the challenges on the voters' behalf. Unfortunately, this means a voter will not be able to *independently* perform auditing. In Chaum's visual crypto scheme [Chaum 2004], the voter needs to choose one of the two transparencies for auditing. However, this implicitly assumes that voters understand how visual cryptography works. In practice, not many voters can grasp the concept of visual cryptography [Karlof et al. 2005]. As explained in Section 3.3, by design, DRE-i is free from these issues. In all three protocols, a universal verifier who has necessary computing expertise is required to verify the audit data published on the bulletin board in one batch operation.

#### 4.4. Comparison with alternative designs

The design of the DRE-i protocol is motivated by the observation that since the touch-screen DRE learns the voter's choice directly and generates random factors for encryption on its own, the involvement of external tallying authorities does not seem strictly necessary for realizing the E2E verifiability. It is worth stressing that there are several ways to construct a "self-enforcing e-voting" protocol and DRE-i is just one of them. While it is beyond the scope of this paper to discuss all possible alternative designs, we will briefly describe one scheme and then compare it with DRE-i.

In order to avoid the involvement of external tallying authorities, one straightforward solution is to adapt the existing TA-based e-voting protocols by merging the functions of the DRE with those of the TAs. For instance, the system may use a single TA and keep the private key in the protected memory of the tamper-resistant module in the DRE machine. All votes are encrypted under the TA's public key on the fly using the standard ElGamal encryption [Kiayias et al. 2006; Clarkson et al. 2008; Adida 2008; Adida et al. 2009] with ciphertext printed on the receipt and also published on the bulletin board. At the end of the election, the DRE machine decrypts the published ciphertext in a verifiable way.

The DRE-i protocol is better than the above alternative design in two main aspects. The first is efficiency. In DRE-i, the ciphertext for the no-vote ( $g^{x_i y_i}$ ) and the yes-vote ( $g^{x_i y_i} \cdot g$ ) consists of a single group element. It takes merely one exponentiation to compute it. By comparison, using the standard ElGamal encryption, it takes two exponentiations to encrypt a vote and the resultant ciphertext consists of two group elements. Second, in DRE-i, all the random factors used in the encryption are fixed before the election with commitment published on the bulletin board, while they are determined on the fly during voting in the alternative design. Our assumption is that the environment in the setup phase is more controllable than that in the field deployment on the election day. Furthermore, the publication of all random public keys ( $g^{x_i}$ ) before the election gives the public an opportunity to verify the distribution of the values, gaining some measure about the randomness. Another practical advantage of pre-fixing the random factors is to allow pre-computing the cryptograms, thus reducing the latency in voting.

## 5. CONCLUSION

E2E verifiable e-voting protocols have been extensively studied in the past twenty years, but the real-world deployment of those protocols has been limited. Our hypothesis is that a key obstacle to the practical deployment is the existing E2E verifiable voting protocols' universal dependence on a set of trustworthy tallying authorities to administer the tallying process. Previous trial experience has shown that implementing such authorities is not an easy task in practice. In this paper, we focus on studying local touch-screen DRE-based elections. First of all, we observe that since the DRE machine learns the voter's choice directly and generates its own random factors for encryption, the involvement of external tallying authorities does not seem strictly necessary for achieving the E2E verifiability. Based on this observation, we propose a self-enforcing e-voting protocol called DRE-i, which provides the same E2E verifiability as previous schemes but without involving any tallying authorities. By comparing DRE-i with related voting systems, we demonstrate encouraging improvements in several aspects, including security, efficiency and usability. This shows that "self-enforcing e-voting", as a new paradigm, has promising potential for further research. In future research, we plan to extend our study to remote e-voting and also to accommodate more complex voting schemes, such as Single Transferable Vote (STV).

## ACKNOWLEDGMENTS

We would like to thank the editors and the anonymous reviewers of USENIX JETS for constructive and pertinent comments. We also thank Ross Anderson, Joseph Bonneau and other members of the security group at the Computer Lab, University of Cambridge, for helpful discussions at the early stage of this research work.

## References

- Ben Adida. 2008. Helios: web-based open-audit voting. In *Proceedings of the 17th USENIX Security Symposium*. USENIX Association, 335–348.
- Ben Adida, Olivier De Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. 2009. Electing a university president using open-audit voting: Analysis of real-world use of Helios. In *Proceedings of the Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE)*. USENIX Association.
- Ben Adida and C Andrew Neff. 2006. Ballot casting assurance. In *Proceedings of the Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE)*. USENIX Association.
- R Michael Alvarez, Gabriel Katz, and Julia Pomares. 2011. The impact of new technologies on voter confidence in Latin America: evidence from e-voting experiments in Argentina and Colombia. *Journal of Information Technology & Politics* 8, 2 (2011), 199–217.
- Ross Anderson. 2008. *Security engineering: a guide to building dependable distributed systems* (second edition ed.). John Wiley & Sons.
- Josh Benaloh. 1987. *Verifiable secret-ballot elections*. Ph.D. Dissertation. Yale University.
- Josh Benaloh. 2007. Ballot casting assurance via voter-initiated poll station auditing. In *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology (EVT)*. USENIX Association.
- Jarrett Blanc. 2007. Challenging the norms and standards of election administration: electronic voting. In *International Foundation For Electoral Systems Report*. 11–19.
- David Chaum. 2004. Secret-ballot receipts: True voter-verifiable elections. *IEEE security & privacy* 2, 1 (2004), 38–47.
- David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L Rivest, Peter YA Ryan, Emily Shen, and Alan T Sherman. 2008a. Scantegrity II: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In *Proceedings of the USENIX/ACCURATE Electronic Voting Workshop (EVT)*. USENIX Association, 1–13.
- David Chaum, Aleks Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, Alan Sherman, and Poorvi Vora. 2008b. Scantegrity: End-to-end voter-verifiable optical-scan voting. *IEEE Security & Privacy* 6, 3 (2008), 40–46.
- David Chaum and Torben Pryds Pedersen. 1993. Transferred cash grows in size. In *Advances in Cryptology – EURO-CRYPT'92 (Lecture Notes in Computer Science)*, Vol. 658. Springer, 390–407.
- Michael R Clarkson, Stephen Chong, and Andrew C Myers. 2008. Civitas: Toward a secure voting system. In *Proceedings of IEEE Symposium on Security and Privacy*. IEEE, 354–368.
- Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. 1994. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology – CRYPTO'94 (Lecture Notes in Computer Science)*, Vol. 839. Springer, 174–187.

- Ronald Cramer, Matthew Franklin, Berry Schoenmakers, and Moti Yung. 1996. Multi-authority secret-ballot elections with linear work. In *Advances in Cryptology – EUROCRYPT’96 (Lecture Notes in Computer Science)*, Vol. 1070. Springer, 72–83.
- Stephanie Delaune, Steve Kremer, and Mark Ryan. 2006. Coercion-resistance and receipt-freeness in electronic voting. In *Proceedings of the 19th IEEE Computer Security Foundations Workshop (CSFW)*. IEEE, 28–39.
- Saghar Estehghari and Yvo Desmedt. 2010. Exploiting the client vulnerabilities in Internet e-voting systems: Hacking Helios 2.0 as an example. In *Proceedings of Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE)*. USENIX Association.
- Amos Fiat and Adi Shamir. 1987. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology – CRYPTO’86 (Lecture Notes in Computer Science)*, Vol. 263. Springer, 186–194.
- Jens Groth. 2004. Efficient maximal privacy in boardroom voting and anonymous broadcast. In *Proceedings of Financial Cryptography (Lecture Notes in Computer Science)*, Vol. 3110. Springer, 90–104.
- Feng Hao, Dylan Clarke, and Carlton Shepherd. 2013. Verifiable classroom voting: Where cryptography meets pedagogy. In *Proceedings of the 21st Security Protocols Workshop (SPW) (Lecture Notes in Computer Science)*, Vol. 8263. Springer, 245–254.
- Feng Hao, Brian Randell, and Dylan Clarke. 2012. Self-enforcing electronic voting. In *Proceedings of the 20th Security Protocols Workshop (SPW) (Lecture Notes in Computer Science)*, Vol. 7622. Springer, 23–31.
- Feng Hao, Peter YA Ryan, and P Zieliński. 2010. Anonymous voting by two-round public discussion. *IET Information Security* 4, 2 (June 2010), 62–67.
- Feng Hao and Piotr Zieliński. 2006. A 2-round anonymous veto protocol. In *Proceedings of the 14th International Workshop on Security Protocols (Lecture Notes in Computer Science)*, Vol. 5087. Springer, 202–211.
- Amir Herzberg, Stanislaw Jarecki, Hugo Krawczyk, and Moti Yung. 1995. Proactive secret sharing or: How to cope with perpetual leakage. In *Advances in Cryptology – CRYPTO’95 (Lecture Notes in Computer Science)*, Vol. 963. Springer, 339–352.
- David Jefferson, A Rubin, Barbara Simons, and David Wagner. 2004. A security analysis of the secure electronic registration and voting experiment (SERVE). <http://servesecurityreport.org>. (2004).
- Chris Karlof, Naveen Sastry, and David Wagner. 2005. Cryptographic Voting Protocols: A Systems Perspective.. In *Proceedings of the 14th USENIX Security Symposium*, Vol. 5. USENIX Association, 33–50.
- Aggelos Kiayias, Michael Korman, and David Walluck. 2006. An Internet voting system supporting user privacy. In *Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC)*. 165–174.
- Aggelos Kiayias and Moti Yung. 2002. Self-tallying elections and perfect ballot secrecy. In *Proceedings of Public Key Cryptography (PKC) (Lecture Notes in Computer Science)*, Vol. 2274. Springer, 141–158.
- Tadayoshi Kohno, Adam Stubblefield, Aviel D Rubin, and Dan S Wallach. 2004. Analysis of an electronic voting system. In *Proceedings of IEEE Symposium on Security and Privacy*. IEEE, 27–40.
- Robert Krimmer, Stefan Triessnig, and Melanie Volkamer. 2007. The development of remote e-voting around the world: A review of roads and directions. In *Proceedings of the 1st International Conference on E-voting and Identity (VOTE-ID)*. 1–15.
- Tomasz Küsters, Ralf Truderung and Andreas Vogt. 2010. Accountability: definition and relationship to verifiability. In *Proceedings of the 17th ACM conference on Computer and communications security (CCS)*. ACM, 526–535.
- Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. 1996. *Handbook of applied cryptography*. CRC press.
- Rebecca T Mercuri. 2001. *Electronic vote tabulation checks and balances*. Ph.D. Dissertation. University of Pennsylvania.
- Wolter Pieters. 2011. How devices transform voting. In *Innovating Government*. Vol. 20. Springer, 439–452.
- Stefan Popoveniuc, John Kelsey, Andrew Regenscheid, and Poorvi Vora. 2010. Performance requirements for end-to-end verifiable elections. In *Proceedings of the 2010 international conference on Electronic voting technology/workshop on trustworthy elections (EVT/WOTE)*. USENIX Association, 1–16.
- Peter YA Ryan, David Bismark, James Heather, Steve Schneider, and Zhe Xia. 2009. Prêt à voter: a voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security* 4, 4 (2009), 662–673.
- Alan T Sherman, Aryya Gangopadhyay, Stephen H Holden, George Karabatis, A Gunes Koru, Chris M Law, Donald F Norris, John Pinkston, Andrew Sears, and Dongsong Zhang. 2006. An examination of vote verification technologies: Findings and experiences from the Maryland Study. In *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop (EVT)*. USENIX Association.
- Michael Steiner, Gene Tsudik, and Michael Waidner. 1996. Diffie-Hellman key distribution extended to group communication. In *Proceedings of the 3rd ACM conference on Computer and communications security (CCS)*. ACM, 31–37.
- Douglas R Stinson. 2006. *Cryptography: theory and practice* (third edition ed.). CRC press.
- VoteHere. 2002. Network Voting System Standards (NVSS). (2002). Public Draft 2.

Scott Wolchok, Eric Wustrow, J Alex Halderman, Hari K Prasad, Arun Kankipati, Sai Krishna Sakhamuri, Vasavya Yagati, and Rop Gonggrijp. 2010. Security analysis of India's electronic voting machines. In *Proceedings of the 17th ACM conference on Computer and communications security (CCS)*. ACM, 1–14.

Richard Wolf. 2008. Voting equipment changes could get messy on Nov. 4. (29 October 2008). Available at [http://www.usatoday.com/news/politics/election2008/2008-10-28-votingequipment\\_N.htm#table](http://www.usatoday.com/news/politics/election2008/2008-10-28-votingequipment_N.htm#table).

## A. FAIL-SAFE DRE-I AND SECURITY PROOFS

We use the same domain parameters,  $(p, q, g)$ , as those defined in Section 2. Assume at the end of the election, a subset  $L$  of ballots are found to be missing (or corrupted) on the bulletin board. To allow the public to verify the tally of the remaining ballots, the DRE publishes  $A = g^{\sum_{i \in L} x_i y_i}$  and proves non-interactively that  $A$  is in the right format without revealing the secrecy of each individual  $g^{x_i y_i}$  term as follows.

- (1) DRE chooses  $r \in_R [1, q - 1]$  and publishes  $X_i = (g^{x_i})^r$  and  $Z_i = (g^{x_i y_i})^r$  for all  $i \in L$ .
- (2) DRE publishes ZKPs of Equality (based on Chaum-Pedersen's technique [Chaum and Pedersen 1993]) for all  $i \in L$  to prove that the discrete logarithm of  $X_i$  with respect to base  $g^{x_i}$  is equal to the discrete logarithm of  $X_j$  with respect to base  $g^{x_j}$ , where  $j$  is the index in  $L$  immediately greater than  $i$ . These ZKPs guarantee that for any  $i, j \in L$  ( $i \neq j$ ),  $X_i = (g^{x_i})^r$  and  $X_j = (g^{x_j})^r$  have the same exponent  $r$ .
- (3) DRE publishes ZKPs of Equality [Chaum and Pedersen 1993] for all  $i \in L$  to prove that  $(X_i, g^{y_i}, Z_i)$  forms a DDH tuple. This is equivalent to proving that the discrete logarithm of  $Z_i$  with respect to base  $g^{y_i}$  is equal to the discrete logarithm of  $X_i$  with respect to base  $g$ . These ZKPs guarantee that for all  $i \in L$ ,  $Z_i = (g^{x_i y_i})^r$  has the same exponent  $r$ .
- (4) DRE publishes a ZKP of Equality [Chaum and Pedersen 1993] to prove that the discrete logarithm of  $\prod_{i \in L} Z_i$  with respect to base  $A$  is equal to the discrete logarithm of an arbitrary  $X_i$  ( $i \in L$ ) with respect to base  $g^{x_i}$ . It suffices to choose  $i$  to be the first index in  $L$ . This ZKP guarantees that  $A$  is indeed represented in the form of  $g^{\sum_{i \in L} x_i y_i}$ .

These published data guarantee that  $A$  is in the correct representation. Therefore  $A$  can be subsequently included into the tallying process to rectify the effects of missing ballots. Among the published data, the ZKP of Equality does not leak anything more than one bit information about the truth of the statement: the two discrete logarithms are equal [Chaum and Pedersen 1993]. However, the process also involves publishing additional data:  $X_i$  and  $Z_i$  for all  $i \in L$ . In the following, we will prove that the  $X_i$  and  $Z_i$  values will not affect the secrecy of each individual  $g^{x_i y_i}$ . In other words, the result in Theorem 3.4 still holds. We consider the extreme case when the available data to an adversary is the maximum: i.e.,  $L$  is a whole set rather than a subset (obviously,  $|L| > 1$ ). We will prove Theorem 3.4 holds even in this extreme case. First of all, we define a variant of the DDH assumption as below.

**ASSUMPTION 2 (3DDH VARIANT).** For a generator  $g$  and randomly chosen  $a, b$ , and  $c$ , given a tuple  $(g, g^a, g^b, g^c, g^{ac}, g^{bc}, g^{abc}, C)$  in which  $C$  is either  $g^{ab}$  or  $g^{ab+1}$ , it is hard to decide whether  $C = g^{ab}$  or  $C = g^{ab+1}$ .

**LEMMA A.1.** Assumption 2 is implied by the DDH assumption.

**PROOF.** First, note that Steiner, Tsudik, and Waidner [Steiner et al. 1996] have proven that DDH is equivalent to the generalized DDH assumption. An instance of the generalized DDH assumption is the three-party DDH assumption (3DDH) which states that for a generator  $g$  and randomly chosen  $a, b$ , and  $c$ , given a tuple  $(g, g^a, g^b, g^c, g^{ab}, g^{ac}, g^{bc})$ , it is hard to distinguish  $g^{abc}$  from random. An equivalent formulation of the 3DDH assumption is as follows: for a generator  $g$  and randomly chosen  $a, b$ , and  $c$ , given a tuple  $(g, g^a, g^b, g^c, g^{ac}, g^{bc}, g^{abc})$ , it is hard to distinguish  $g^{ab}$  from random. This can be easily seen by considering  $g^c$  as the generator in the original formulation.

Now we prove that the latter formulation of 3DDH implies Assumption 2. Similar to the proof of Lemma 3.1, consider the following tuples:



$$\begin{aligned} & (g, g^a, g^b, g^c, g^{ac}, g^{bc}, g^{abc}, g^{ab}), \\ & (g, g^a, g^b, g^c, g^{ac}, g^{bc}, g^{abc}, R), \\ & (g, g^a, g^b, g^c, g^{ac}, g^{bc}, g^{abc}, R'g), \quad \text{and} \\ & (g, g^a, g^b, g^c, g^{ac}, g^{bc}, g^{abc}, g^{ab}g), \end{aligned}$$

for random  $a, b, c, R$ , and  $R'$ . 3DDH guarantees that the first and second tuples are indistinguishable. The second and third tuples have the exact same distribution and hence are indistinguishable. 3DDH also guarantees that the third and fourth tuples are indistinguishable. Hence, the first and fourth tuples, i.e.  $(g, g^a, g^b, g^c, g^{ac}, g^{bc}, g^{abc}, g^{ab})$  and  $(g, g^a, g^b, g^c, g^{ac}, g^{bc}, g^{abc}, g^{ab+1})$  are indistinguishable.  $\square$

**LEMMA A.2.** *Consider two failsafe DRE- $i$  elections in which all the votes are exactly the same except for two votes  $v_i$  and  $v_j$  which are swapped between the two elections. Under Assumption 2, the bare bulletin boards of the above two elections are indistinguishable to an adversary that determines an arbitrary number of the votes other than  $v_i$  and  $v_j$ .*

**PROOF.** Let us assume w.l.o.g. that  $i < j$ . If  $v_i = v_j$ , the lemma holds trivially. In the following we give a proof for  $v_i \neq v_j$ .

Let us assume there is an adversary  $\mathcal{A}$  that first chooses an arbitrary number of the votes other than  $v_i$  and  $v_j$ , and eventually distinguishes the two elections. We construct an algorithm  $\mathcal{S}$  that uses  $\mathcal{A}$  to break Assumption 2.

Given a tuple  $(g, g^a, g^b, g^c, g^{ac}, g^{bc}, g^{abc}, C)$ , where  $C$  equals either  $g^{ab}$  or  $g^{ab+1}$ ,  $\mathcal{S}$  sets up the bulletin board with the generator  $g$  as follows. Let  $I = \{1, \dots, n\} \setminus \{i, j\}$ .

$g^{x_k}$  and  $g^{y_k}$  are set up in the same way as the proof of Lemma 3.3. First,  $\mathcal{S}$  chooses  $n-2$  random values  $x_k$  for all  $k \in I$ .  $\mathcal{S}$  sets  $g^{x_i} \leftarrow g^a$ ,  $g^{x_j} \leftarrow g^b$ , and calculates  $g^{x_k}$  for all  $k \in I$ . Note that we implicitly have  $x_i = a$  and  $x_j = b$ . Let  $s_1 = \sum_{k < i} x_k$ ,  $s_2 = \sum_{i < k < j} x_k$ , and  $s_3 = \sum_{k > j} x_k$ .  $\mathcal{S}$  also calculates  $s_1, s_2$ , and  $s_3$  and then computes  $\sigma_i = s_1 - s_2 - s_3$  and  $\sigma_j = s_1 + s_2 - s_3$ .

Now given all  $g^{x_k}$ , all  $g^{y_k}$  can be computed accordingly. Note that we implicitly have:

$$\begin{aligned} y_i &= \sum_{k < i} x_k - \sum_{k > i} x_k = s_1 - (s_2 + b + s_3) = \sigma_i - b \\ y_j &= \sum_{k < j} x_k - \sum_{k > j} x_k = (s_1 + a + s_2) - s_3 = \sigma_j + a \end{aligned}$$

Next,  $\mathcal{S}$  simulates  $g^{x_k r}$  and  $g^{y_k r}$  as follows. It sets  $g^{x_i r} \leftarrow g^{ac}$  and  $g^{x_j r} \leftarrow g^{bc}$ ; that is, we implicitly have  $r = c$ . For all  $k \in I$ , it sets  $g^{x_k r} \leftarrow (g^c)^{x_k}$ . Then it sets

$$g^{x_i y_i r} \leftarrow (g^{ac})^{\sigma_i} / g^{abc} = g^{a(\sigma_i - b)c} \quad \text{and} \quad g^{x_j y_j r} \leftarrow (g^{bc})^{\sigma_j} \cdot g^{abc} = g^{b(\sigma_j + a)c}.$$

In general, for any  $k = 1, \dots, n$ , we define  $\sigma_k = \sum_{\substack{\ell < k \\ \ell \neq i, j}} x_\ell - \sum_{\substack{\ell > k \\ \ell \neq i, j}} x_\ell$ . Now we have:

$$\forall k \in I: \quad y_k = \sum_{\ell < k} x_\ell - \sum_{\ell > k} x_\ell = \pm a \pm b + \sum_{\substack{\ell < k \\ \ell \neq i, j}} x_\ell - \sum_{\substack{\ell > k \\ \ell \neq i, j}} x_\ell = \pm a \pm b + \sigma_k,$$

where depending on  $k$ , we have either a plus or a minus sign in front of  $a$  and  $b$  and  $\sigma_k$  is known. Hence  $\{g^{x_k y_k r}\}_{k \in I}$  can be simulated as

$$g^{x_k y_k r} \leftarrow \left( g^{\pm ac} g^{\pm bc} (g^c)^{\sigma_k} \right)^{x_k} = g^{x_k (\pm a \pm b + \sigma_k)c}.$$

Table VI. The simulated bare bulletin board in the proof of Lemma A.2

$k$	$g^{x_k}$	$g^{y_k}$	$g^{x_k r}$	$g^{x_k y_k r}$	$g^{x_k y_k} g^{v_k}$
1	$g^{x_1}$	$1 / \prod_{k>1} g^{y_k}$	$(g^c)^{x_1}$	$((g^c)^{\sigma_1} / (g^{ac} g^{bc}))^{x_1}$	$(g^{x_1})^{y_1} g^{v_1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$i$	$g^a$	$\prod_{k<i} g^{x_k} / \prod_{k>i} g^{y_k}$	$g^{ac}$	$(g^{ac})^{\sigma_i} / g^{abc}$	$(g^a)^{\sigma_i} \cdot g/C$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$j$	$g^b$	$\prod_{k<j} g^{x_k} / \prod_{k>j} g^{y_k}$	$g^{bc}$	$(g^{bc})^{\sigma_j} \cdot g^{abc}$	$(g^b)^{\sigma_j} \cdot C$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n$	$g^{x_n}$	$\prod_{k<n} g^{x_k}$	$(g^c)^{x_n}$	$(g^{ac} g^{bc} (g^c)^{\sigma_n})^{x_n}$	$(g^{x_n})^{y_n} g^{v_n}$

$\mathcal{S}$  sets up the last column of the bare bulletin board similar to the proof of Lemma 3.3.  $\mathcal{A}$  chooses a set of votes  $\{v_k\}_{k \in I_{\mathcal{A}}}$  for the set of indexes  $I_{\mathcal{A}} \subseteq I$ . Let us consider some arbitrary set of votes  $\{v_k\}_{k \in I \setminus I_{\mathcal{A}}}$ .  $\mathcal{S}$  can calculate  $g^{x_k y_k}$  for all  $k \in I$ , since it knows  $x_k$  and  $g^{y_k}$ . Hence, it can calculate  $g^{x_k y_k} g^{v_k}$  for all  $k \in I$ . For  $k = i, j$ ,  $\mathcal{S}$  sets

$$g^{x_i y_i} g^{v_i} \leftarrow (g^a)^{\sigma_i} \cdot g/C \quad \text{and} \quad g^{x_j y_j} g^{v_j} \leftarrow (g^b)^{\sigma_j} \cdot C.$$

Now the calculation of the entire bare bulletin board is complete. Table VI shows the simulated bare bulletin board.

In the case that  $C = g^{ab}$ , we have:

$$g^{x_i y_i} g^{v_i} \leftarrow (g^a)^{\sigma_i} \cdot g/C = (g^a)^{\sigma_i} \cdot g/g^{ab} = g^{a(\sigma_i - b)} g = g^{x_i y_i} g \quad \text{and}$$

$$g^{x_j y_j} g^{v_j} \leftarrow (g^b)^{\sigma_j} \cdot C = (g^b)^{\sigma_j} \cdot g^{ab} = g^{b(\sigma_j + a)} = g^{x_j y_j},$$

which means that in our bare bulletin board  $v_i = 1$  and  $v_j = 0$ .

In the case that  $C = g^{ab+1}$ , we have:

$$g^{x_i y_i} g^{v_i} \leftarrow (g^a)^{\sigma_i} \cdot g/C = (g^a)^{\sigma_i} \cdot g/g^{ab+1} = g^{a(\sigma_i - b)} = g^{x_i y_i} \quad \text{and}$$

$$g^{x_j y_j} g^{v_j} \leftarrow (g^b)^{\sigma_j} \cdot C = (g^b)^{\sigma_j} \cdot g^{ab+1} = g^{b(\sigma_j + a)} g = g^{x_j y_j} g,$$

which means that in our bare bulletin board  $v_i = 0$  and  $v_j = 1$ .

$\mathcal{S}$  then gives  $\mathcal{A}$  the constructed bare bulletin board as input. If  $\mathcal{A}$  is able to distinguish which of the above two cases the given bare bulletin board corresponds to,  $\mathcal{S}$  will be able to successfully distinguish the two cases for  $C$  and hence break Assumption 2.  $\square$

**THEOREM A.3 (MAIN THEOREM).** *Under the DDH assumption and that the ZKP primitives used in the protocol are secure, the failsafe DRE- $i$  bulletin board does not reveal anything about the secrecy of the votes other than the tally of non-adversarial votes to an adversary that determines an arbitrary number of votes.*

**PROOF.** Similar to the proof of Theorem 3.4, whereas now we rely on Lemmas A.1 and A.2 instead.  $\square$

A corollary of the above theorem can be stated as below for a passive adversary that does not determine any votes, but only observes the bulletin board.

**COROLLARY A.4 (PRIVACY AGAINST PASSIVE ADVERSARIES).** *Under the assumptions that DDH is intractable and the ZKP primitives used in the protocol are secure, the failsafe DRE-i bulletin board does not reveal anything about the secrecy of the votes other than the tally of the votes to a passive adversary.*

## **Usability of Voter Verifiable, End-to-end Voting Systems: Baseline Data for Helios, Prêt à Voter, and Scantegrity II**

**Claudia Z. Acemyan<sup>1</sup>, Philip Kortum<sup>1</sup>, Michael D. Byrne<sup>1,2</sup>, Dan S. Wallach<sup>2</sup>**

<sup>1</sup>Department of Psychology, Rice University

<sup>2</sup>Department of Computer Science, Rice University

6100 Main Street, MS-25

Houston, TX 77005 USA

{claudiaz, pkortum, byrne}@rice.edu and dwallach@cs.rice.edu

### **ABSTRACT**

In response to voting security concerns, security researchers have developed tamper-resistant, voter verifiable voting methods. These end-to-end voting systems are unique because they give voters the option to both verify the system is working properly and to check that their votes have been recorded after leaving the polling place. While these methods solve many of the security problems surrounding voting with traditional methods, the systems' added complexity might adversely impact their usability. This paper presents an experiment assessing the usability of Helios, Prêt à Voter, and Scantegrity II. Overall, the tested systems were exceptionally difficult to use. Data revealed that success rates of voters casting ballots on these systems were extraordinarily low. Specifically, only 58% of ballots were successfully cast across all three systems. There were reliable differences in voting completion times across the three methods, and these times were much slower than previously tested voting technologies. Subjective usability ratings differed across the systems, with satisfaction being generally low, but highest for Helios. Vote verification completion rates were even lower than those for vote casting. There were no reliable differences in ballot verification times across the three methods, but there were differences in satisfaction levels, with satisfaction being lowest for Helios. These usability findings—especially the extremely low vote casting completion rates—highlight that it is not enough for a system to be secure; every system must also be usable.

### **INTRODUCTION**

For centuries there has been a desire for auditability in elections. In mid-19<sup>th</sup> century America, groups of voters stood in public venues and called out their ballot choices to the election clerks, while a judge tallied the votes (Jones, 2001). The advantage of this voting method was that anyone could listen to the vocal expression of preferences and keep their own vote count, which prevented practices like ballot box stuffing. While this oral voting method may have increased the accuracy of vote counting, voters' desire for privacy was not addressed, enabling bribery and coercion. In response, during the late 1800s, voting jurisdictions began to introduce the use of the secret, Australian ballots that listed all the candidates for the same office on the same sheet of paper (which was issued to voters at the polling station) and guaranteed voters privacy in preparing ballots inside a booth (Brent, 2006). This voting system ensured that voters prepared their own ballot expressing their intent while preserving anonymity. Yet this voting method was not perfect; there was not a means to audit the election—leaving a long-standing tension between auditability and privacy in elections.

### **e2e Voting Systems**

So that cast ballots can be both auditable and anonymous, which would ultimately improve the integrity of elections, voting security researchers have developed secure, voter verifiable systems, also known as end-to-end (e2e) voting systems (e.g., Adida, 2008; Carback et al., 2010; Chaum et al., 2010; Clarkson, 2008; Ryan et al., 2009). e2e systems are voting methods that aim for ballots to be cast as voters intend and counted as cast. To make sure these systems are functioning as they should, they are designed so that both voters and observers can audit, or verify, various aspects of the voting method—all while preserving voter privacy.

How do these e2e systems work? To protect votes from malicious attacks, cryptographic protocols and auditing mechanisms are used. The cryptographic methods make it very difficult to undetectably attack and/or alter the e2e systems so that election outcomes would be impacted. Then, with the ability for voters and observers to audit the system, people are given a means to make sure the system is working as it should—from making certain that intended selections are the actual votes cast to checking that the ballots are accurately counted, resulting in a fair, accurate election. In order to protect the identity and preferences of the voter, information that could identify the voter is never associated with the ballot. Instead, e2e systems use a unique ballot identifier (such as a code associated with each ballot), allowing a voter to find and identify their own ballot while preventing others from being able to tell that the specific ballot belongs to that individual. In addition, when a voter goes through the verification process to check that their ballot was cast and recorded, their actual ballot selections are never revealed. Rather, the voter may be shown another type of information that confirms that their ballot selections are recorded without disclosing the actual selections.

Examples of e2e voting systems include Helios (Adida, 2008), Prêt à Voter (Ryan et al., 2009), and Scantegrity II (Chaum et al., 2008). These three systems have been selected to be representative examples of voter verifiable systems for several reasons. First, they are largely accepted and discussed as secure voting methods within the voting research community. Furthermore, they represent a spectrum of the different solution types that have been proposed for use in polling stations (it has been suggested that Helios can be modified and adapted for use at polling sites in order to prevent coercion). Helios is a web-based system and an exemplar of Benaloh-style schemes (Benaloh, 2006). Prêt à Voter (PaV) is a simple, novel, paper-based scheme with many variants that are being considered for use in various elections all over the world. Scantegrity II is another paper-based scheme that incorporates the traditional paper bubble ballot. All three voting systems have been used, or will be used, in actual elections: Helios was used in the presidential election at the Université Catholique de Louvain, Belgium (Adida et al., 2009), International Association for Cryptologic Research's board of directors election (IACR, n.d.), and Princeton Undergraduate Elections (see [princeton.heliosvoting.org](http://princeton.heliosvoting.org)). PaV has been used in student elections in both Luxembourg and Surrey (P. Ryan, personal communication, April 3, 2014), and it will be used in the November 2014 Victorian State elections (Burton et al., 2012). Scantegrity II was used in the November 2009 municipal election in Takoma Park, Maryland (Carback et al., 2010).

#### *Helios*

Helios is a web-based, open-audit voting system (Adida, 2008; Adida et al., 2009) utilizing peer-reviewed cryptographic techniques. From a security standpoint, system highlights include browser-based encryption, homomorphic tallying, distributed decryption across multiple trustees, user authentication by email address, election-specific passwords, and vote casting assurance through various levels of auditing.

From the voter's standpoint, Helios appears to be similar to direct recording electronic voting systems (DREs) like VoteBox (Sandler, et al, 2008). Instances of the user interface can be seen in Appendix 1. The following outlines the vote casting process from the voter's perspective (the exact steps have the potential to vary from voter to voter, hence the following are potential procedures): 1) The voter logs into their email account to obtain the election's website address (this information can also be disseminated through other methods). 2) After navigating to the election's Helios Voting Booth webpage, the voter reads through the voting system instructions and clicks "start" to begin voting. 3) The voter completes the ballot one race at a time by checking the box next to the desired candidate or proposition and then clicking next/proceed to move onto the next screen. 4) The voter reviews his or her ballot and then clicks the "confirm choices and encrypt ballot" button. 5) The voter records his or her smart ballot tracker by printing it out and proceeds to submission. 6) The voter logs in with their email address to verify their eligibility to vote. 7) The voter casts the ballot associated with their smart ballot tracker. 8) The voter views a screen indicating their vote has been successfully cast.

For a voter to verify their vote, or check that it was in fact cast in the election, the following sequence is typical: 1) In the user's inbox, open and view an email from the Helios Voting Administrator. The e-mail indicates that their vote has been successfully cast and displays a link where the ballot is archived. 2) The voter clicks on the ballot archive link. 3) The voter views a screen that says "Cast Vote" along with their smart ballot tracker. The voter clicks on details and views the code associated with the ballot, which can be used on an auditing page to verify that their ballot is encrypted correctly. 4) The voter returns to the election home page and clicks on "Votes and Ballots." 5) The voter observes on the Voter and Ballot Tracking Center page that their smart ballot tracker is shown within the list of cast votes.

#### *Prêt à Voter*

The next system, Prêt à Voter (PaV), inspired by Chaum's (2004) visual cryptographic scheme, is a voting system that allows voters to vote with paper forms (with randomly ordered races and selections for each race), which can be physically modified to then serve as an encrypted ballot. This voting method is auditable at numerous phases by both voters and teams of auditors (Ryan et al., 2009). The system is flexible in that it allows different encryption schemes and cryptographic mechanisms to be used as needed.

PaV was intended to provide voters with a simple, familiar voter experience. Images of this study's voting instructions, ballot, receipt, and vote verification pages can be found in Appendix 2.

To vote with the PaV system, the voter follows these typical steps: 1) A sealed envelope enclosing a paper ballot is given to the voter. The voter opens the envelope and finds an instruction sheet and cards that make up the ballot. 2) To mark their selections on the ballot cards, a cross (x) is marked in the right hand box next to the name of the candidate or proposition that the voter wants to select. 3) After completing the ballot, the voter detaches the candidates lists from their selections or marks. 4) The candidates lists are shredded. 5) The voter walks over to the vote casting station and feeds the voting slips into the scanner. 6) The voting slips are placed in the ballot box. 7) The voter takes a printed receipt, which shows images of the scanned voting slips along with the website and ballot verification code needed to confirm that they voted.

For a voter to verify their vote using PaV, the voter might typically perform the following sequence on a computer or mobile device: 1) Navigate to the election verification website, which is printed on their receipt. 2) Enter the ballot verification code on the home page and submit it. 3) View the vote validation page that confirms the entered verification code is valid. This page also

displays images of every ballot card—thereby displaying every selection on every card (without any candidates lists) that makes up their ballot.

### *Scantegrity II*

The third method, Scantegrity II, is an optical scan voting system that enables a voter to vote with a paper bubble ballot, enhanced by traceable confirmation codes that can be revealed by invisible ink decoder pens (Chaum et al., 2008). This voting system can be audited by voters or any other interested party.

Scantegrity II was developed so that voters could still use a familiar voting technology—an optical scan bubble ballot that they already have experience using. Images of the paper bubble ballot and other voting system materials used in this study can be found in Appendix 3.

To cast a vote using the Scantegrity II voting method, a voter would typically do the following: 1) Read the instructions on both the ballot and separate vote verification sheet. 2) Use the special marking device to make ballot selections—and consequently reveal codes—by filling in the appropriate bubbles. 3) Record on the separate vote verification sheet the revealed confirmation codes found inside each marked bubble. Also record on this sheet the ballot ID / online verification number that is found on the bottom right corner of the ballot. 4) Walk over to the ballot casting station to scan in the ballot and have it then placed in the ballot box. 5) Hand the vote verification sheet to the polling station official so that they can stamp “Cast Ballot” on it. 6) Choose whether or not to keep their verification sheet.

To verify the votes, a voter may perform the following sequence at their home or office: 1) Navigate to the election’s vote verification web page. 2) Enter their unique online verification number associated with their ballot. 3) View a confirmation webpage that says the ballot has been cast and processed. This page also displays the online validation code along with a list of the voter’s confirmation codes, with each code corresponding to a ballot selection.

### **Understanding the Usability of e2e Voting Systems**

As can be seen from the vote casting and vote verification procedures, the three e2e systems are complex from the standpoint of the voter. Many of the processes required to use the systems are both long and novel in the context of voting. This is of concern because voters already have difficulty voting with standard paper ballots due to design deficiencies like insufficient instructions and confusing ballot designs (Norden et al., 2008). If additional e2e mechanisms are then laid on top of these problems, this raised the question of whether or not voters’ abilities to cast their votes will be further degraded. If people cannot use the system to vote, then voters will likely be disenfranchised and election outcomes might be changed—tremendous threats to democracy. Furthermore, if people are not able to verify that their ballot has been cast because the system is too hard to use, then the system is not auditable—leaving room for inaccuracy and corruption. Consequently, voting researchers need to understand the usability of each system and how it compares to other voting technologies.

System usability is defined as the capability of a range of users to be able to easily and effectively fulfill a specified range of tasks within specified environmental scenarios (Shackel, 1991). In the context of voting, usability might be thought of as whether or not voters can use a voting method to successfully cast their votes. Per ISO standard 9241-11 (1998), there are three suggested measurements of usability: effectiveness, efficiency and satisfaction. As established in previous voting usability research (Byrne et al., 2007; Laskowski et al., 2004), effectiveness addresses whether or not voters are able to select, without error, the candidate or proposition for which they

intend to vote. One way to measure effectiveness is by calculating error rates. Efficiency concerns the amount of resources required of a voter to attempt achieving his or her goal. This variable can be measured by calculating task completion times, or the amount of time it takes to vote or verify a vote. The third measure, satisfaction, is defined as the voter's subjective perceptions of a voting system after using it—such as how hard or easy it is to vote using the method. Satisfaction can be measured with a standardized instrument like the System Usability Scale, or SUS (Brooke, 1996).

The only way to know if e2e systems are usable is to empirically test them. While other studies have reported on the usability of select e2e systems (Carback et al., 2010; Karayumak, 2011; Weber et al., 2009, Winckler et al., 2009), none have experimentally evaluated the voting methods along all three suggested measurements outlined by both ISO standard 9241-11 and the 2004 NIST report on voting system usability (Laskowski et al., 2004).

To address this lacuna, this study tested the usability of the three e2e voting systems presented above: Helios, Prêt à Voter, and Scantegrity II. When applicable, the same materials and protocols were used from the previous voting studies conducted by Rice University's human factors voting laboratories (e.g., Byrne et al., 2007; Campbell et al., 2009; Campbell et al., 2011; Everett, 2007; Everett et al., 2008; Holmes & Kortum, 2013) to allow for comparison of usability findings across different voting technologies. The goals of this research project were to understand whether voters can use these e2e voting methods to cast and verify their votes, identify system attributes that might be preventing voters from fulfilling their goals of vote casting and verifying, and help us to make recommendations that might enhance the design and implementation of e2e systems.

## **METHODS**

### **Participants**

Thirty-seven participants who were U.S. citizens and 18 years or older (the minimum age to vote in the U.S.) were recruited through an online advertisement in Houston, Texas. They were paid \$40 for participating in the study. The mean age was 37.1 years, with a median of 35 and a range of 21 to 64. There were 22 male and 15 female participants. Participants were African American (14, 38%), Caucasian (10, 27%), Mexican American / Chicano (4, 11%), Hispanic / Latino (4, 11%), and other ethnicities (5, 13%). As for the participants' educational background, 2 (5%) had completed high school or the GED, 23 (62%) completed some college or an associate's degree, 8 (22%) were awarded a bachelor's degree or equivalent, and 4 (11%) held a post-graduate degree. English was the native language of 36 of these participants. All had self-reported normal or corrected-to-normal vision. Participants rated their computer expertise on a scale from 1 to 10, with one being novice and 10 being expert; the mean was 8.2 with a range of 5 to 10. 33 participants had voted in at least one national election, with an average of 3.8 and a range of 0 to 21. Participants had, on average, voted in 5.1 state and local elections. This is a diverse and representative sample of real voters.

### **Design**

A within-subjects design was used, in which every participant used three different voting methods. The within-subjects study design increased the statistical power of the analysis such that the sample size of 37 was more than adequate to detect even small effects. The three voting systems used in this experiment were Helios, Prêt à Voter, and Scantegrity II. Each participant voted with all three methods. All possible orders of presentation were used, and subjects were randomly assigned an order.



So that voters knew for whom they should vote, they were given a list of candidates and propositions. Their list was either primarily Republican and contained 85% Republican candidates, or it was primarily Democratic with 85% being Democratic candidates. Both lists had “yes” votes for four propositions and “no” votes for two. These two lists were the same as those used in our previous studies. Participants were randomly assigned one of the two slates.

Per the ISO 9241-11 definition of usability (ISO, 1998), there were three main dependent variables: errors (effectiveness), completion time (efficiency), and subjective usability (satisfaction). Three types of errors were included in the effectiveness measure. First, we measured the inability to either cast a ballot and/or later verify votes. For example, if a participant completed a ballot but never cast it by scanning it, then this was counted as an error with PaV and Scantegrity II. In Helios, if a voter encrypted his or her ballot but never continued on to verify their eligibility to vote (by logging in with their email account)—an action that is required at this point in the voting process in order to move onto the actual vote casting step, then this would be counted as a failure to cast. Second, we recorded per-race errors, which are defined as deviations on the voter’s ballots from the list of candidates and propositions given to the voter, which they were instructed to select. A per-contest error rate for each ballot was computed for every participant. Third, overall ballot errors were measured. Overall ballot errors are defined as a ballot with at least one deviation from the list of candidates and propositions given to the voter. For example, whether a voter selected one wrong candidate or ten wrong candidates, the ballot would be classified as having errors on it.

To measure efficiency, voting and verification completion times were used. Both voting and vote verification times were measured with a stopwatch. The stopwatch was started after the experimenter said the participant could begin, and it was stopped when the participant indicated that they were finished with their task.

The System Usability Scale was used to measure satisfaction. The SUS contains ten subscales. Each subscale is a 5-point Likert scale that measures an aspect of usability. The ratings for each subscale are combined to yield a single usability score ranging from 0 to 100, with lower scores being associated with lower subjective usability.

Data were also collected on other factors such as technologies used to vote in previous elections, computer experience, perceptions of voting security, and preferred voting technology.

For each e2e system, the dependent measures described above were collected for both the vote casting portion of the system (i.e., the procedures the voter must go through in order to make their selections on a ballot and successfully cast the ballot), as well as the vote verification portion of the system (i.e., the procedures required of the voter to be able to check that their votes were cast and included in the final election tally). The two portions of the system were examined separately since vote verification is an optional procedure not required to cast a ballot and have it be counted. This study did not explore the usability of the optional auditing processes associated with the systems.

### **Procedures**

The study began with participants giving their informed consent. They were then read instructions for the experiment. Subjects were instructed to vote on all three ballots according to their list of candidates and propositions. Because verification is neither currently an option in U.S. elections, nor required to cast a vote with e2e systems, voters were specifically told that they would be asked

to verify their vote at the end of the voting process, and that they should take whatever steps were necessary to insure that they could perform this verification step. Participants then voted with one of the three voting methods (order was counterbalanced across participants, all orders used), each in its own room to prevent confusion as to which equipment was associated with each voting system. After voting on a system, the participants immediately completed the System Usability Scale. When completing the instrument, participants were specifically instructed to evaluate the *voting* system they had just used. Next, participants verified their vote using the same system and completed another SUS, being explicitly instructed to evaluate only the *verification* system they just used. They then went through this process for the remaining two systems. At the end of the experiment, participants completed a final survey packet that was composed of 49 questions. The survey covered topics like demographics, computer expertise, previous voting experience, security, voting method comparisons, voting method instructions, and vote verification. Last, participants were debriefed, compensated, and thanked for their time.

We used the modified form of the System Usability Scale as presented in Bangor et al. (2008) to assess subjective usability or satisfaction. In this version of the SUS, the word “cumbersome” is replaced with “awkward.” We also replaced the word “system” with the words “voting system” or “voting method,” and “verification system” or “verification method” as appropriate. We made this particular change based on user feedback from our pilot study’s subjects. Altering the SUS in this way has been shown to have no impact on the scale’s reliability (Sauro, 2011).

It should be noted that the participants’ desktops were mirrored to a monitor that only the experimenter could view in another part of the room. Mirroring the monitors was intended to aid the experimenter in observing the participant’s actions in an unobtrusive fashion. Mirrored monitors also allowed the experimenter to score the errors on Helios’ ballot in real time and determine if voters verified their votes across all three systems.

### **Materials**

For all three systems, the following hardware was used: The computers were Dell Optiplex desktops with 17” monitors. The scanners were VuPoint Solution Magic Wands; these scanners were selected because they would automatically feed and scan sheets of paper inserted by the user. The shredders used were Amazon Basics 8 or 12-sheet automatic shredders. The printers used were the HP Deskjet 1000 (Helios) and the HP LaserJet Pro Laser Printer (PaV), both of which are single function printers. All computers had Windows XP operating systems and Google Chrome version 32 as the default web browser. This web browser was selected because it was compatible with all voting and verification systems tested in this study. The only icons on the computers’ desktops were the hard drive, trashcan, and Google Chrome.

Candidates and propositions on the ballots were those used in our previous experiments (e.g., Byrne et al., 2007; Everett et al., 2008). The candidates’ names had been randomly generated through online software. The ballot was comprised of 21 races, which included both national and county contests, and six propositions. The length and composition of the ballot was originally designed to reflect the national average number of races. The format and layout of each system’s ballot followed the criteria outlined by the system developers in published papers.

The Helios voting system and election was set up and run through Helios’ website at [vote.heliosvoting.org](http://vote.heliosvoting.org) during the winter of 2013-2014. A Gmail login provided to the participant was used to obtain Helios voting instructions, access the election link, confirm eligibility/identity before casting the ballot, and/or view the confirmation email sent after ballot casting. See Appendix 1 for the study materials used in association with this voting system.

Since PaV had not been previously developed to be used in an election with numerous races (as is the case in the United States), our team developed the system based on published papers about PaV (e.g., Lundin & Ryan, 2008; Ryan et al., 2009; Ryan & Peacock, 2010; Ryan & Schneider, 2006), the PaV website (Prêt à Voter, n.d.), and in consultation with Peter Ryan, who first created the system. It should be noted that the security mechanisms were not implemented in the system. Nevertheless, from the voter's perspective, the system appeared to operate as a fully functional, secure system. See Appendix 2 for system materials.

This study's implementation of Scantegrity II was heavily based on materials used in the 2009 Takoma Park, Maryland election, in which voters used the system to elect the mayor and city council members (Carback et al., 2010). We also referred to published articles about the system and corresponded through email with Aleks Essex, a researcher who has direct experience with the implementation. When aspects of the system that might have potential to impact usability were not specified, best practices in human factors were followed. Also, when possible, every effort was made to keep system properties (such as font) constant across systems. Like PaV, this system was not a fully functional prototype from a security perspective. Instead, it appeared to be fully functional from the voter's perspective. See Appendix 3 for Scantegrity II's materials.

## RESULTS

There were no differences in the findings based on whether participants were told to vote for mostly Republicans or mostly Democrats according to their directed voting list, so we treated this as a single condition. There were also no differences in the efficiency, effectiveness, and satisfaction findings based on whether or not participants were able to cast a vote or later verify a vote. This was also treated as one condition. The analysis was a repeated measures ANOVA unless otherwise specified. *p*-values were adjusted by Greenhouse-Geisser (G-G) correction when appropriate. FDR adjustments to post-hoc tests were performed when necessary.

### Vote Casting

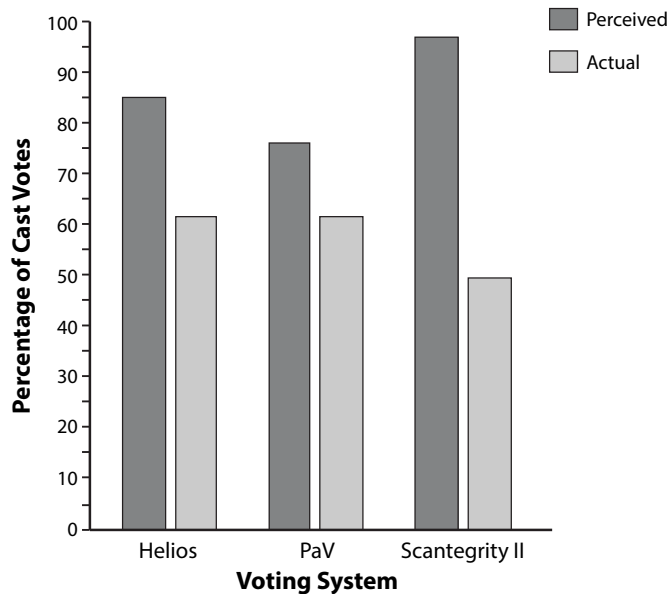
#### *Effectiveness*

Figure 1 shows the number of voters who thought they cast a vote with each system versus the number of actual cast votes. As can be seen, a reliably higher percentage of voters *thought* they had cast a vote that would be counted in election totals than the percentage of ballots that they *actually* cast, (tested with binomial linear mixed model,  $z = 4.42$ ,  $p < .001$ ). The interaction between these two variables across voting systems was not reliable. These completion rate findings are extremely troubling. If the tested e2e voting systems are used in a real election, on a large scale, high percentages of voters might not be able to vote—resulting in disastrous outcomes. These failure-to-cast findings are especially unacceptable when many of the other systems tested in our lab produced 100% ballot casting completion rates (e.g., Byrne et al., 2007).

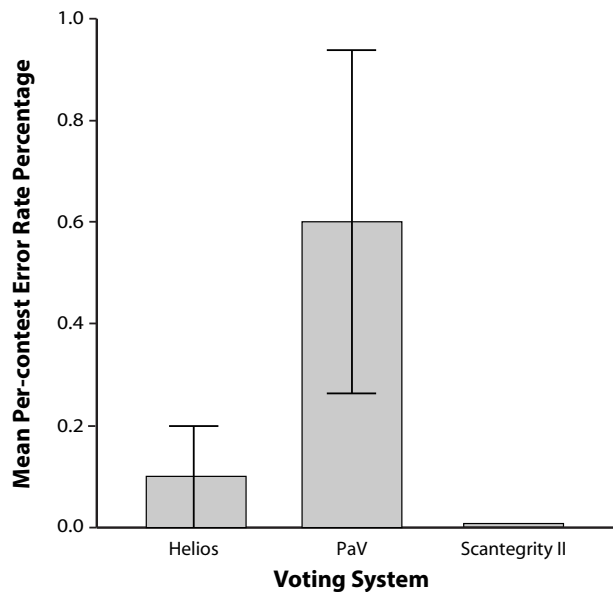
Per-contest error rates as a function of system can be seen in Figure 2. There was no reliable evidence for an effect of system type on these errors,  $F(1.1, 40.9) = 2.70$ ,  $MSE = 0.00$ ,  $p = .104$ ,  $\eta^2 = .09$ . In this regard, e2e systems seem to be performing better than previously tested voting systems that had error rates ranging from less than 0.5% to about 3.5% (Byrne et al., 2007). With that being said, this potential advantage over other voting technologies is moot if voters cannot cast votes at reasonable rates.

Table 1 shows the frequency of error-containing ballots by voting system. Overall, 5 of the 111 (5%) ballots collected contained at least one error. Again, this error rate is lower than those previously reported (see Byrne et al., 2007). Based on both the per-contest error rates and error

rates by ballot, voters using e2e systems make few errors selecting candidates and propositions on their ballots.



**Figure 1.** Percentage of cast ballots as a function of voting system, with different colored bars representing perceived and actual cast votes



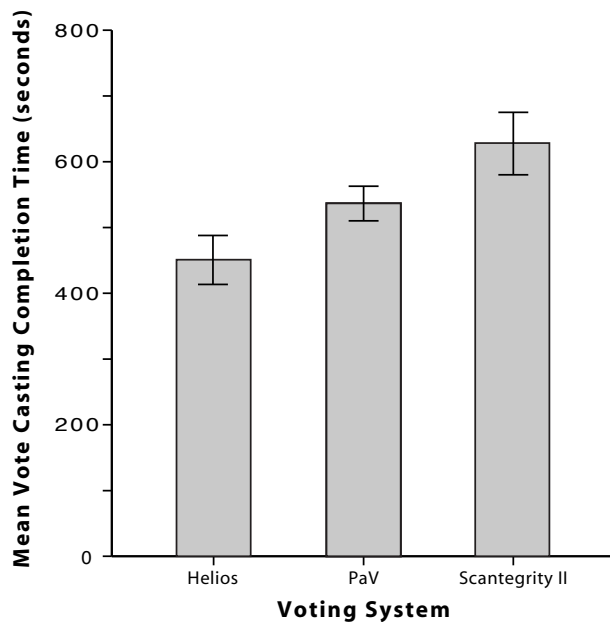
**Figure 2.** Mean per-contest error rate percentage as a function of voting system type, with error bars depicting the standard error of the mean

**Table 1.** The number and percent of ballots with one or more errors as a function of voting system type

	Helios	PaV	Scantegrity II
<b>Number of Ballots with Errors</b>	1 (3%)	4 (11%)	0 (0%)

*Efficiency*

Average ballot completion time as a function of voting system is presented in Figure 3. As can be seen, there are differences in voting times across the systems,  $F(2, 72) = 8.45$ ,  $MSE = 34,457$ ,  $p = .001$ ,  $\eta^2 = .23$ . Pairwise tests revealed all three means were reliably different. Participants took the least amount of time to vote with Helios and the most amount of time to vote with Scantegrity II. In prior research, ballot completion time is generally not sensitive to voting technology. Average completion time for the identical ballot using arrow ballot, bubble ballot, punch card, and lever machine voting methods is approximately 231 seconds (Byrne et al., 2007) and 290 seconds across sequential DRE, direct DRE, bubble ballot, lever machine, and punch card systems (Everett et al., 2008). Thus, the e2e systems impose a substantial time cost on voters.

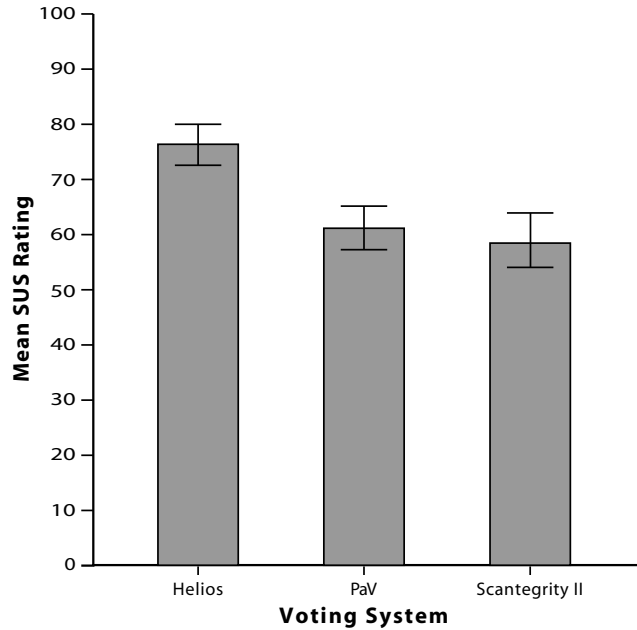


**Figure 3.** Mean vote casting completion time as a function of voting system, with error bars depicting the standard error of the mean

*Satisfaction*

As can be seen in Figure 4, SUS ratings (out of 100 possible points) differ across the three e2e voting systems,  $F(2, 72) = 5.28$ ,  $MSE = 624$ ,  $p = .007$ ,  $\eta^2 = .13$ . Pairwise *t*-tests revealed that participants were reliably more satisfied with the usability of Helios, but there was not a statistically reliable difference in satisfaction ratings between PaV and Scantegrity II. When compared to previously tested voting methods, these SUS scores are comparable or lower than those previously seen (Byrne et al., 2007). Using the assessment of fitness for use scale (based on

the SUS score) proposed by Bangor, Kortum and Miller (2009), Helios would be judged as “acceptable,” while PaV and Scantegrity II would be on the low end of “marginal acceptability.” Based on all of these SUS findings, voters’ satisfaction with using Helios was relatively good, but their satisfaction with using the other two systems was between poor and good—suggesting that there is room for improvement in future system iterations.



**Figure 4.** Mean SUS rating as a function of voting system, with error bars depicting the standard error of the mean

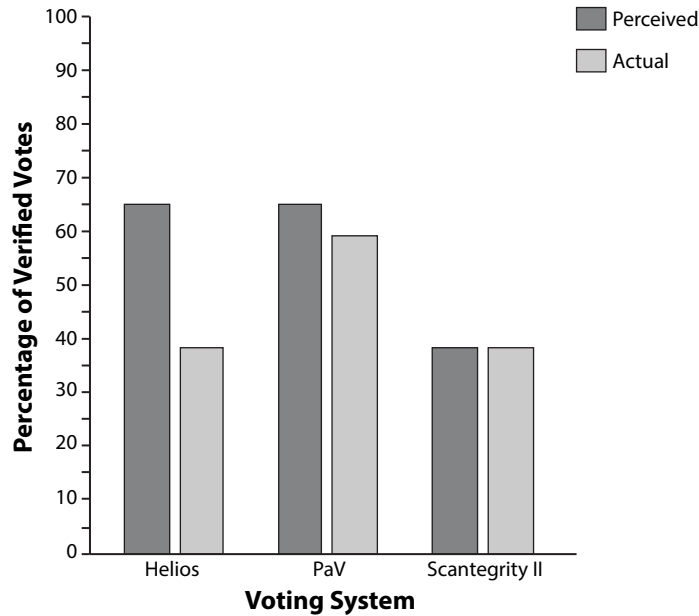
### Vote Verification

#### *Effectiveness*

Figure 5 shows the number of participants who were able to actually verify their vote through any means versus those who thought they verified as a function of system type. There was no reliable effect of system or difference between perceived versus actual completion rates. However, these vote verification task completion rates are lower than those for vote casting (again, tested via binomial linear mixed model,  $z = 2.17, p = .030$ ).

With Helios, 16 (43%) voters performed any type of vote verification action. Of these, only 8 (50%) recorded their smart ballot tracker, which allows them to identify their particular vote in the online vote center. Two of the 16 participants verified by viewing the verification email sent to them after voting. The rest of the subjects verified by viewing their information on the Helios election website, keeping in mind that many did not have a recorded smart ballot tracker to which they could refer. With Scantegrity II, 14 (38%) voters performed some type of vote verification. Of these, only nine attempted to record all 27 vote verification codes; only a *single* person wrote down all 27 correctly. Based on these results, for both Helios and Scantegrity II participants engaged in a wide range of behaviors when they tried to check that their vote was cast in the mock elections. PaV was designed so that the verification output required to check on the ballot was automatically given to voters upon casting their ballots, and there was only one way in which they

could check on their ballots, so more specific findings on verification actions are not reported for the system.



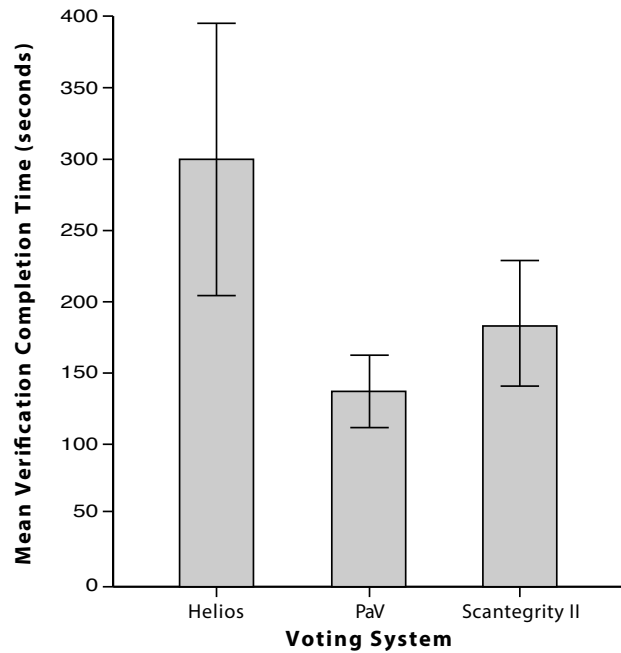
**Figure 5.** Percentage of verified votes as a function of voting system, with different colored bars representing perceived and actual verified votes

*Efficiency*

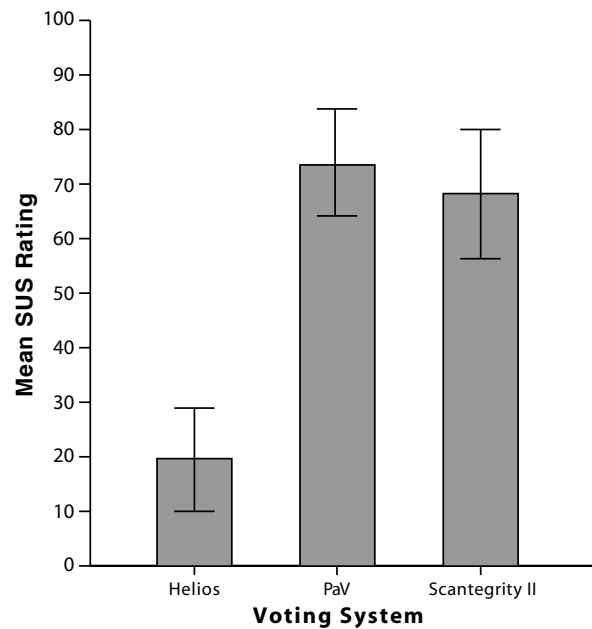
Results for vote verification time as a function of voting system are presented in Figure 6. The effect of voting system was suggestive but not statistically reliable,  $F(1.2, 7.2) = 3.74$ ,  $MSE = 21,559$ ,  $p = .089$ ,  $\eta^2 = .38$ . It should be noted that the amount of time it takes someone to *verify* their vote with these e2e voting systems is similar to the amount of time it takes to *vote* on previously tested voting technologies (Byrne et al., 2007).

*Satisfaction*

Figure 7 depicts the mean SUS score as a function of system type. The effect of voting system was reliable,  $F(2, 12) = 7.86$ ,  $MSE = 792$ ,  $p = .007$ ,  $\eta^2 = .57$ . Pairwise *t*-tests indicated that Helios was rated lower than PaV on the subjective usability measure; there was not any evidence to support other statistically reliable differences. Using the assessment of fitness for use scale (Bangor et al., 2009), Helios would be judged as being “not acceptable,” Scantegrity II would be on the high end of “marginal,” and PaV would be classified as “good.” To summarize these findings, Helios’ verification system had a staggeringly low subjective usability rating, emphasizing how bad participants thought of the system’s usability. Participants did rate PaV higher (that is, that they thought PaV was easier to use).



**Figure 6.** Mean verification completion time as a function of voting system, with error bars depicting the standard error of the mean



**Figure 7.** Mean SUS rating for the vote verification process as a function of voting system, with error bars representing the standard error of the mean



## DISCUSSION

Generally, all of the tested e2e voting systems appear to have momentous usability issues based just on the high failure-to-cast rates. Perhaps more troubling, however, is the fact that many of the participants in this study *thought* they cast a vote, but actually did not. These findings would have huge implications in a real election. Since they believe they did in fact vote, they would not even know to tell someone that they could not cast a vote to receive assistance or notify officials that there might be usability problems. As for the voters who recognize they cannot vote, they might seek help or they might give up. Even if they are able to eventually cast a vote after receiving direction, they might choose not to vote in the future, and thus the e2e systems would disenfranchise voters.

The low success rates observed in the vote verification part of the systems are also troublesome. If voters cannot check on their ballot after voting, then fewer people will be able to check that the system is working properly. The voter might also have lower confidence in the system since they know the verification feature is available, but they were not able to use it for some reason. Even if a voter is able to verify that his or her vote was cast, it might lead to frustration levels that are associated with future system avoidance, meaning—again—there will be fewer people to check on the integrity of the system. One potentially unintended consequence of these verification systems is that it adds another opportunity for errors to be committed. If the voters write down their verification information incorrectly (a smart ballot tracker in the case of Helios or a selection's confirmation code with Scantegrity II) then they might think their vote was lost, thrown out, or not recorded correctly. If the voter then reports to an election official that something is wrong, a new set of serious problems emerge: election officials and voters might think the election results are incorrect, when in fact they are correct. If widespread, this kind of simple and foreseeable failure could lead to a general lack of confidence in the results among the "average" voter who tried to verify their vote, but failed. These are all serious ramifications—highlighting that it is not enough for a system to be secure. Every system must also be usable.

### Why are these systems failing?

It is clear that while the e2e mechanisms may significantly enhance the security of these voting systems, the enhancements come at the cost of usability. The additional and unfamiliar procedures impact the very essence of the voting process—the ability to cast a vote—and do so in ways that cause many users to not even be aware that they have failed. We believe that there are several general design choices that led to the results reported here, yet each of these can be overcome with design modifications and additional research efforts.

#### *1) Security Isn't Invisible*

All of the tested e2e voting systems function in a way that require users to be an active part of the security process. These additional steps likely lead to increased cognitive load for the user, and that increased load can lead to failures. In contrast, an ideal security mechanism requires no such additional effort on the part of the user. In novice parlance, "it just happens." The user is neither required to take action nor even know that there is enhanced security implemented on his behalf. For example, banks encrypt their web-based transactions, but the user does not take part in enabling or executing these additional safety measures.

#### *2) Tested e2e Systems Do Not Model Current Systems to the Greatest Degree Possible*

Many of the observed usability difficulties in this study can likely be attributed to designs that work differently than users expect. Many participants were experienced with voting and had seen previous (albeit, different) implementations of what a voting system "should" look like and how it

“should” behave. For the most part, the tested e2e systems deviated from these expectations significantly, leaving users confused. In this confusion, participants might have recalled their previous experience with voting systems, and then used that to guide their interactions. Since their previously used voting systems do not work in the same way as e2e voting systems, referring to previous experience inevitably led to decreases in performance and the commission of errors where the users’ prior voting model and the system’s actual function did not match. This may explain why Helios had higher SUS ratings than PaV and Scantegrity II. Many participants verbally expressed that they liked using the computer to vote since they already use them daily—in other words, they got to use a platform with which they were familiar. Of the three systems, Helios also requires the least amount of unfamiliar, novel procedures. Essentially, the voter only has to interact with a series of webpages to vote. In contrast, with PaV voters have to tear their completed ballot in half, shred a portion of it, and then scan what is leftover into a scanner. Scantegrity II is similarly unique, requiring voters to use decoder pens, record revealed invisible ink codes, and then scan in their ballot. Deviations from the norm can hurt performance and user assessment of that system, which is reflected in our results. Furthermore, PaV and Scantegrity both require that candidate order be randomized, which violates the expectations of most voters and does not conform to election laws in most U.S. jurisdictions.

Even though voters have never seen or interacted with systems like these before, it should not be argued that high rates of failure to cast a vote or to verify a vote are to be expected—hence being acceptable in a system deployed for use. This argument can be countered in two ways. First, completion rates for two previously tested experimental voting systems—IVR and mobile vote—do not suffer from this phenomenon (Holmes & Kortum, 2013; Campbell et al., in press). Second, and more importantly, voting should be considered a walk-up-and-use activity. If a voter only votes in national elections, then there are four years between each interaction a voter has with a particular system, and learning retention is poor under infrequent exposures. Voters must be able to use the system with near 100% success with little or no experience or training.

### *3) Verification Output Is Not Automated, So Users Make Mistakes*

Verification of a vote is a new feature of these systems, so this probably led to some of the system problems like not being able to verify or recognize that their vote had been verified. However, the benefits derived from this feature are so central to these enhanced security systems that more needs to be done to assist voters in the successful completion of this step. As noted, one of the great difficulties users faced is that they either failed to understand that they needed to record additional information to verify, or the additional labor involved dissuaded them from making the effort. Further, even if voters understood and wanted to perform these steps, the likelihood of committing errors in this step was high. Providing assistance to the voter, such as automated output of the ballot ID (which PaV did) or security codes might have made this step more tenable from the voter’s standpoint.

### *4) Insufficient User Instructions*

Because these e2e system are both relatively new and place additional cognitive burdens on the users, enhanced instruction may be required. This does not necessarily mean giving the voters long, detailed instructions for use at each station, as these were often ignored or skimmed in the systems tested here. It does mean providing specific, clear helping instructions at critical junctures in the process. Instructions should never be a substitute for good design, but occasionally, good inline dialogue can mitigate design features that are crucial to the systems operation. This lack of inline instruction may have been why subjective usability was lowest for Helios. Helios provided instructions in the beginning on how to vote, but after casting a ballot, the system did not tell the voter how they could follow up by verifying to be assured that their vote was handled correctly.

### *5) Voting Systems Were Not Specified in Detail*

One of the things learned quickly as our team tried to construct these systems is that while the security mechanisms were well-specified by the researchers who imagined them, not every system specification was defined. This is understandable, as the papers we used to model e2e systems described the security and general functioning of the system, not every single operational user interface detail. However, anyone (like a county clerk) who wanted to implement such a system would be left to devise their own best practices for all the omitted details, and this could lead to a wide range of outcomes depending on the implementation. The devil is always in the details, and this is especially true for complex systems such as these. It also points to the need for enhanced collaboration between security researchers and human factors specialists when developing such systems.

### **Where do we go from here?**

Despite the usability problems associated with the tested systems, one must keep in mind that they have the potential to be both more secure and more accurate than traditional voting systems once the systems are usable by everyone. Incorporating human factors research and development methods during active system development would be a critical part of ensuring that these types of systems are developed with the user in mind

There are numerous questions that future research should address. For example, are people with disabilities able to use the voter verifiable systems? If not, what can be done so that they can easily and quickly vote? Are the auditing portions of the system usable? When a voter verifies their vote with a system like Scantegrity II or PaV that displays their unique codes or images of their ballot, how accurate are voters? In other words, would people actually catch errors? How do voters report concerns about their verified votes? All three systems are designed to allow voters to check that things are working properly. But if they are not, what do voters do? By answering questions like these, the systems will be able to be further improved and the relationship between security and usability will be understood in more detail.

### **CONCLUSION**

The data from this study serves as a reference point for future research and discussions about the usability of voter verifiable voting systems. It also enables e2e systems to be compared to other voting systems that have been previously tested or will be tested in the future. With that being said, this study only begins to answer basic research questions surrounding these new systems, while highlighting many avenues for future studies.

### **ACKNOWLEDGEMENTS**

This research was supported in part by the National Institute of Standards and Technology under grant #60NANB12D249. The views and conclusions expressed are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of NIST, the U.S. government, or any other organization.

## REFERENCES

- Adida, B. (2008). Helios: Web-based open-audit voting. *Proceedings of the 17<sup>th</sup> USENIX Security Symposium, USA, 17*, 335-348.
- Adida, B., De Marneffe, O., Pereira, O., & Quisquater, J. J. (2009). Electing a university president using open-audit voting: Analysis of real-world use of Helios. *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections, USA, 18*.
- Bangor, A., Kortum, P.T., Miller, J.T. (2008). An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction, 24*(6), 574-594.
- Benaloh, J. (2006). Simple verifiable elections. *Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop, USA, 15*.
- Brent, P. (2006). The Australian ballot: Not the secret ballot. *Australian Journal of Political Science, 41*(1), 39-50.
- Brooke, J. (1996). SUS: A 'quick and dirty' usability scale. In P.W. Jordan, B. Thomas, B.A. Weerdmeester, & I.L. McClland (Eds.), *Usability Evaluation in Industry* (pp. 189-194). Bristol: Taylor & Francis.
- Byrne, M. D., Greene, K. G., & Everett, S. P. (2007). Usability of voting systems: Baseline data for paper, punchcards, and lever machines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM* (pp. 171-180).
- Burton, C., Culnane, C., Heather, J., Peacock, T., Ryan, P. Y., Schneider, S., ... & Xia, Z. (2012, July). Using Prêt a Voter in Victorian State elections. *Proceedings of the 2012 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections, USA, 21*.
- Campbell, B. A., & Byrne, M. D. (2009). Now do voters notice review screen anomalies? A look at voting system usability. *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections, USA, 18*.
- Campbell, B. A., Tossell, C. C., Byrne, M. D., & Kortum, P. (2011, September). Voting on a Smartphone Evaluating the Usability of an Optimized Voting System for Handheld Mobile Devices. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting: Vol. 55*(1). *Human Factors and Ergonomics Society* (pp. 1100-1104).
- Campbell, B. A., Tossell, C. C., Byrne, M. D., Kortum, P. (in press). Toward more usable electronic voting: Testing the usability of a smartphone voting system. In *Human Factors*.
- Carback, R., Chaum, D., Clark, J., Conway, J., Essex, A., Herrnson, P.S., . . . Vora, P.L. (2010). Scantegrity II Municipal Election at Takoma Park: The first e2e binding governmental election with ballot privacy. *Proceedings of the 19th USENIX Security Symposium, USA, 19*.
- Chain voting prevented by new ballots. (1931, August 27). *The Gettysburg Times*, p. 1.
- Chaum, D. (2004). Secret ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy, 2*(1), 38-47.
- Chaum, D., Carback, R., Clark, J., Essex, A., Popoveniuc, S., Rivest, R. L., ... & Sherman, A. T. (2008). Scantegrity II: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. *Proceedings of EVT '08, USA*.
- Chaum, D., Jakobsson, M., Rivest, R. L., Ryan, P. Y., Benaloh, J., & Kutyłowski, M. (Eds.). (2010). *Lecture Notes in Computer Science: Vol. 6000. Towards Trustworthy Elections: New Directions in Electronic Voting*. New York, NY: Springer.
- Clarkson, M. R., Chong, S. N., & Myers, A. C. (2008). Civitas: Toward a secure voting system. In *Proceedings of the 2008 IEEE Symposium on Security & Privacy. IEEE Computer Society* (pp. 354-368).
- Everett, S. P. (2007). *The Usability of Electronic Voting Machines and How Votes Can Be Changed Without Detection* (Doctoral dissertation, Rice University). Retrieved from <http://chil.rice.edu/alumni/petersos/EverettDissertation.pdf>
- Everett, S., Greene, K., Byrne, M., Wallach, D., Derr, K., Sandler, D., & Torous, T. (2008). Electronic voting machines versus traditional methods: Improved preference, similar

- performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM (pp. 883-892).
- Holmes, D., & Kortum, P. (2013). Vote-By-Phone: Usability Evaluation of an IVR Voting System. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting: Vol. 57(1)*. Human Factors and Ergonomics Society (pp. 1308-1312).
- IACR. (n.d.). *Should the IACR Use E-Voting for Its Elections?* Retrieved from <http://www.iacr.org/elections/eVoting/>
- ISO. (1998). *Ergonomic requirements for office work with visual display terminal (VDT's)–Part 11: Guidance on usability* (ISO 9241-11(E)). Geneva, Switzerland.
- Jones, D.W. (2001). A brief illustrated history of voting. *Voting and Elections Web Pages*. Retrieved from <http://homepage.cs.uiowa.edu/~jones/voting/pictures>
- Karayumak, F., Kauer, M., Olembo, M., Volk, T., & Vokamer, M. (2011). User study of the improved helios voting system interfaces. In *2011 1<sup>st</sup> Workshop on Socio-Technical Aspects in Security and Trust (STAST)*. IEEE Computer Society (pp. 37-44).
- Laskowski, S.J., Autry, M., Cugini, J., Killam, W., & Yen, J. (2004). Improving the usability and accessibility of voting systems and products. Washington: D.C.: National Institute of Standards and Technology. Retrieved from <http://ucdwww.user.openhosting.com/files/NISTHFReport.pdf>
- Lundin, D., & Ryan, P.Y. (2008). Human readable paper verification of Prêt à Voter. In S. Jajodia & J. Lopez (Eds.), *Computer Security – ESORICS 2008: Proceedings of the 13<sup>th</sup> European Symposium on Research in Computer Security, Malaga, Spain, October 6-8, 2008* (pp. 379-395). Berlin, Germany: Springer Berlin Heidelberg.
- Masnack, M. (2008). Guy Who Insists E-Voting Machines Work Fine... Demonstrates They Don't. *Tech Dirt*. Retrieved from <http://www.techdirt.com/articles/20081029/0131342676.shtml>
- Norden, L., Kimball, D., Quesenbery, W. & Chen, M. (2008). *Better Ballots*. New York: Brennan Center for Justice. Retrieved from <https://www.supportthevoter.gov/files/2013/08/Better-Ballots-Brennan-Center.pdf>
- Prêt à Voter. (n.d.). Retrieved from <http://www.pretavoter.com>
- Ryan, P. Y., Bismark, D., Heather, J., Schneider, S., & Xia, Z. (2009). Prêt à voter: a voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security*, 4(4), 662-673.
- Ryan, P.Y., & Peacock, T. (2010). A threat analysis of Prêt à Voter. In D. Chaum, M. Jakobsson, R.L. Rivest, P.Y. Ryan, J. Benaloh, & M. Kutylowski, (Eds.), *Lecture Notes in Computer Science: Vol. 6000. Towards Trustworthy Elections: New Directions in Electronic Voting* (pp. 200-215). New York, NY: Springer.
- Ryan, P.Y., & Schneider, S.A. (2006). Prêt à Voter with re-encryption mixes. In D. Gollmann, J. Meier, & A. Sabelfeld (Eds.), *Computer Security – ESORICS 2006: Proceedings of the 11<sup>th</sup> European Symposium on Research in Computer Security, Hamburg, Germany, September 18-20, 2006* (pp. 313-326). Berlin, Germany: Springer Berlin Heidelberg.
- Sandler, D., Derr, K., & Wallach, D. S. (2008). VoteBox: A Tamper-evident, Verifiable Electronic Voting System. *Proceedings of the 17th USENIX Security Symposium, USA, 4*.
- Sauro, J. (2011, February 2). Measuring usability with the system usability scale (SUS) [Web log post]. Retrieved from <https://www.measuringusability.com/sus.php>
- Shackel, B. (1991). Usability-context, framework, definition, design and evaluation. In *Human Factors for Informatics Usability* (pp. 21-37). New York, NY: Cambridge University Press.
- Weber, J., & Hengartner, U. (2009). *Usability study of the open audit voting system Helios*. Retrieved from <http://www.jannaweber.com/wpcontent/uploads/2009/09/858Helios.pdf>
- Winckler, M., Bernhaupt, R., Palanque, P., Lundin, D., Ryan, P., Alberdi E., & Strigini, L. (2009). *Assessing the usability of open verifiable e-voting systems: a trial with the system Pret a Voter*. Retrieved from <http://www.irit.fr/~Marco.Winckler/publications/2009-ICEGOV.pdf>

## Appendix 1--Helios Voting System Study Materials

**General Election  
Harris County, Texas  
November 8, 2016**

To participate in this election, you will need to use the internet. For voting instructions, please go to: **mail.google.com**

Login to Gmail using the following information:

**Username:** xraychicken  
**Password:** suitandtie

Figure A1.1. Study instructions for the Helios mock-election

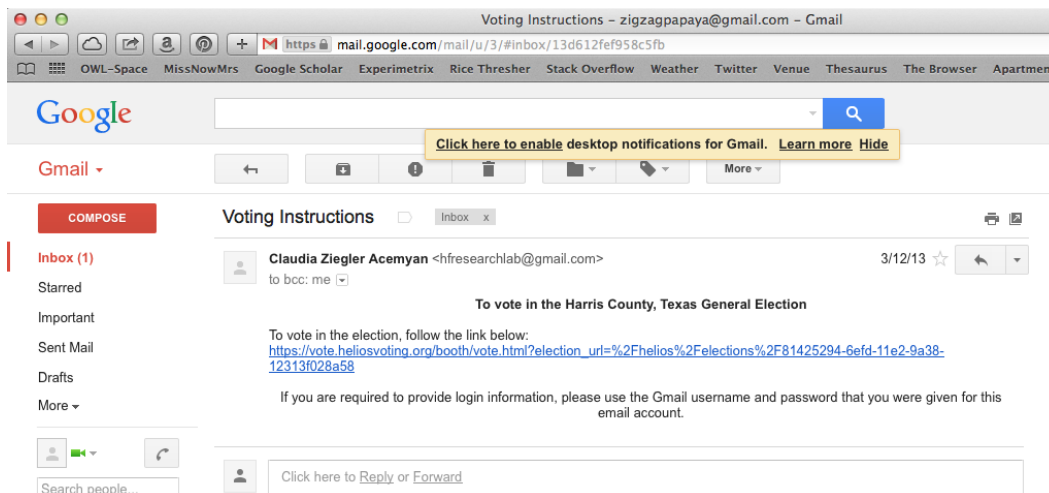


Figure A1.2. Screenshot of the emailed instructions and link to the Helios election

## Appendix 1--Helios Voting System Study Materials

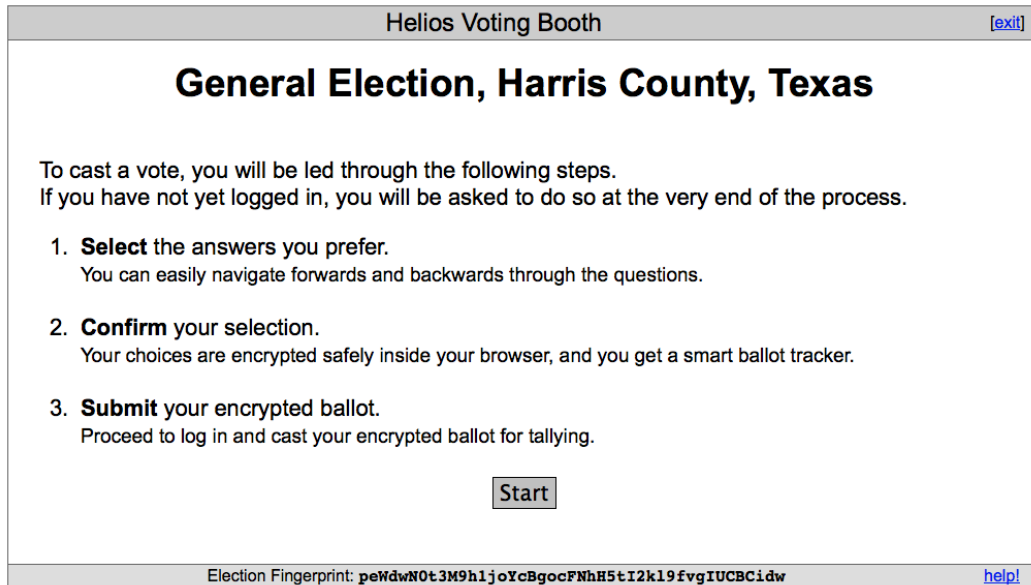


Figure A1.3. Screenshot of the Helios Voting Booth instructions

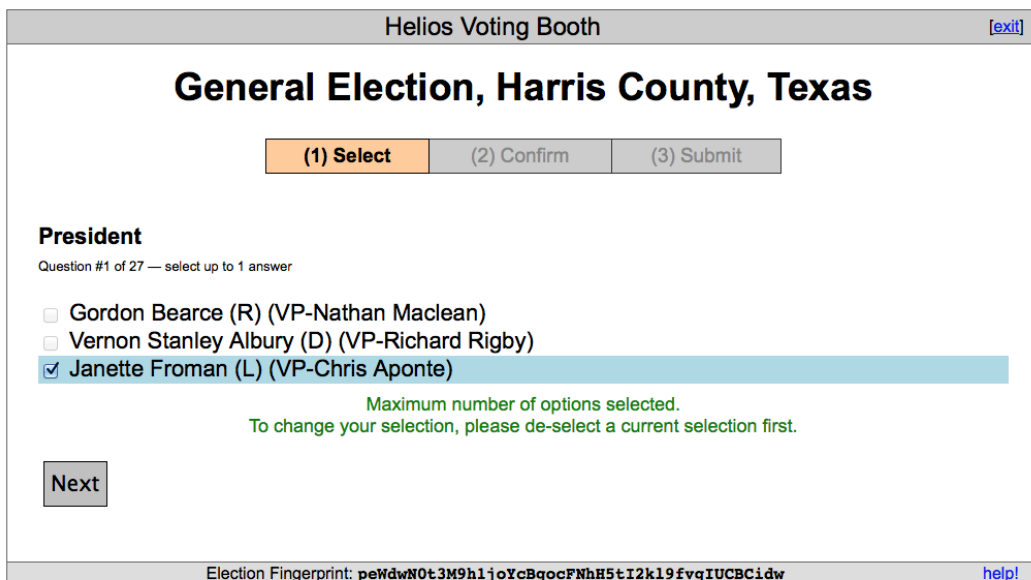


Figure A1.4. Screenshot of the presidential race on the Helios Ballot

## Appendix 1--Helios Voting System Study Materials

**Question #25:** Proposition 4: Shall there be an amendment to the Texas revised statutes concerning renewable energy standards for large providers of retail electric service, and, in connection therewith, defining eligible renewable energy resources to include solar, wind, geothermal, small hydroelectricity, and hydrogen fuel cells; requiring that a percentage of retail electricity sales be derived from renewable sources, beginning with 3% in the year 2007 and increasing to 10% by 2015; requiring utilities to offer consumers a rebate of \$2.00 per watt and other incentives for solar electric generation; providing incentives for utilities to invest in renewable energy resources that provide net economic benefits to customers; limiting the retail rate impact of renewable energy resources to 50 cents per month for residential customers; requiring public utilities commission rules to establish major aspects of the measure; prohibiting utilities from using condemnation or eminent domain to acquire land for generating facilities used to meet the standards; requiring utilities with requirements contracts to address shortfalls from the standards; and specifying election procedures by which the customers of a utility may opt out of the requirements of this amendment?

[\[update\]](#)

**Question #26:** Proposition 5: Shall there be an amendment to the Texas constitution concerning election day voter registration, and, in connection therewith, allowing an eligible citizen to register and vote on any day that a vote may be cast in any election beginning on January 1, 2007; specifying election day voter registration locations; specifying that an eligible citizen who registers to vote on election day shall register in person and present a current and valid Texas driver's license or state identification card or other approved documentation; and directing the Texas general assembly, in implementing election day voter registration, to adopt necessary protections against election fraud?

[\[update\]](#)

**Question #27:** Proposition 6: Shall the Charter of Harris County concerning the powers of the City Council be amended in regard to the sale of city-owned property, to require Council approval for the sale of personal property valued at \$500,000 or more, and to clarify language requiring Council approval of any sale of real property?

[\[update\]](#)

**Confirm Choices and Encrypt Ballot**

Election Fingerprint: `peWdwN0t3M9h1joYcBgocFNhH5tI2k19fvgIUCBCidw` [help!](#)

Figure A1.5. Screenshot of the Helios review screen


Helios Voting About Code Docs FAQ Privacy Help


# General Election, Harris County, Texas — Submit your Vote


We have received, **but not yet recorded**, your encrypted ballot.  
Your smart ballot tracker is:

**cIzxampJ0vDs iJB0/vZySlwU8Iv+GV2rdwZGDxw3i4k**

Now, we need you to log in, so we can verify your eligibility.

 google

 facebook

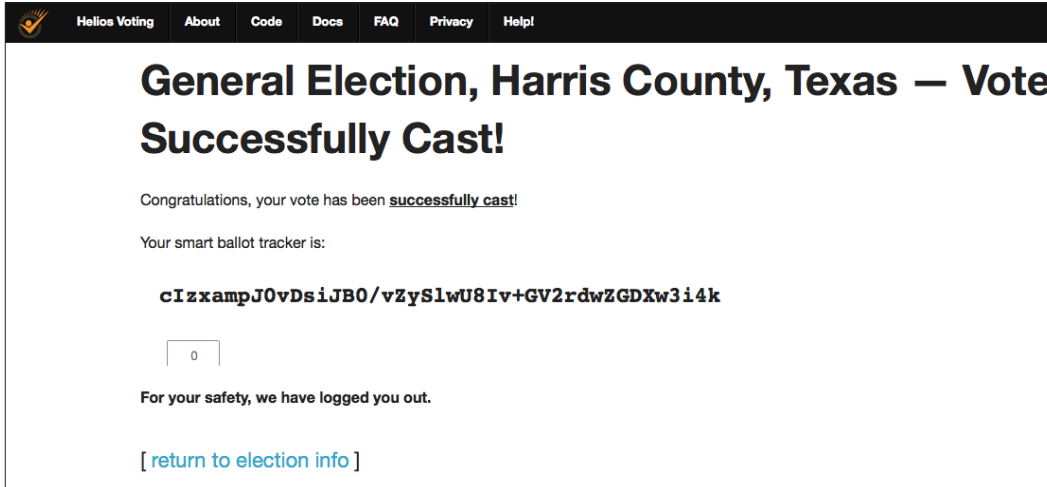
 yahoo

Don't worry, we'll remember your ballot while you log in.

Figure A1.6. Screenshot of one Helios vote submission page



## Appendix 1--Helios Voting System Study Materials

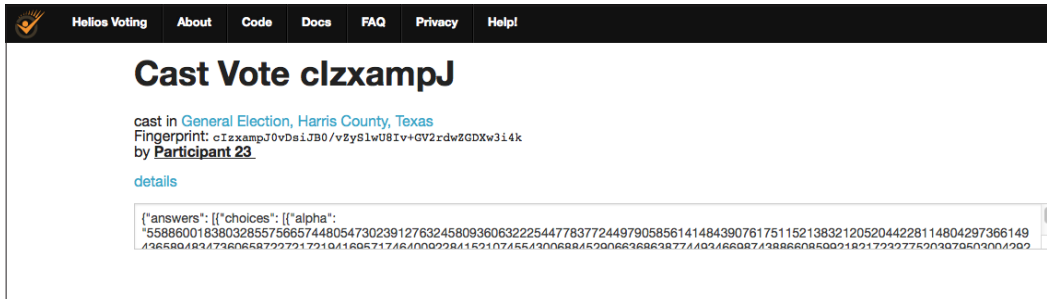


**Figure A1.7.** Screenshot of the Helios cast vote confirmation page, which is shown at the end of the voting process

red monkey	x/hVL7Yg/F4cQ6t12d4WfdmaV4wDUKNcTkqEWPP7CPA <a href="#">[view]</a>
Participant 83	5vWdkhyE849pE9sC51zY+CBURhMlhccW7WGYN6IBR2A <a href="#">[view]</a>
Participant 74	X1OeO8ilgiZ7q50jjdwNO2FNi7ltXEfhjb8LABwRhSA <a href="#">[view]</a>
Participant 80	V9eoWPeDjL75wPOyr6qH4dyKZzhdNNYo8qcbUj7pMsI <a href="#">[view]</a>
Human Factors	0WduOUzrVb+QbVhMzuLLI+ENQcXdv3pbjWIjoXcadh4 <a href="#">[view]</a>
Participant 10	DjxiWadCamDcgG8qMA+bNWTZ3aDiEDgW71Bx01aLaw8 <a href="#">[view]</a>
Participant 11	7Xt6BNmnGZwy2FatxHgWAeVdEEQo9kQ5usIkwPmfYC8 <a href="#">[view]</a>
Participant 14	—
Participant 17	gpyTOMwLAJ1+QIWeHSZ1T05GtRu0w509gPvHk/831W4 <a href="#">[view]</a>
Participant 18	IpGY4e+mT7Qfz0WNiwzj5RPWgt2JcS/y+8YG9BOJtXo <a href="#">[view]</a>
Participant 1	LnTLOEz0D5TlWvCWtAY1wCjNnzBPAlxxrTplygHxaqI <a href="#">[view]</a>
Participant 4	WLselSilhPcvGq96EFFLVgI8Br6whgT+qaykLH+LXSM <a href="#">[view]</a>
Human Factors	b0pUYLZ7EdLORbJilksjjwz9/Vnr1Jxs96kUaj03NqA <a href="#">[view]</a>
yellow owl	/WMZXEAOH1Itj7bLJZoLOf12R5jrI36aKliWwhqmVLI <a href="#">[view]</a>
Participant 23	cIzxampJ0vDs iJB0/vZySlwU8Iv+GV2rdwZGDxw3i4k <a href="#">[view]</a>

**Figure A1.8.** Screenshot of Helios' Voters and Ballot Tracking Center

## Appendix 1--Helios Voting System Study Materials



**Figure A1.9.** Screenshot of a voter’s archived ballot (accessed by voter through the emailed cast ballot confirmation link)

## Appendix 2--Pret a Voter Voting System Study Materials

**General Election Ballot**  
**Harris County, Texas**  
**November 8, 2016**

**INSTRUCTIONS TO VOTERS**

**1. Mark a cross (x) in the right hand box next to the name of the candidate you wish to vote for.** For an example, see the completed sample ballot below. Use only the marking device provided or a number 2 pencil. Please note that this ballot has multiple cards. If you make a mistake, don't hesitate to ask for a new ballot. If you erase or make other marks, your vote may not count.

Cathy	<input type="checkbox"/>	<b>Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.</b> <span style="float: right;">Vote Verification Code</span>
Eliot	<input type="checkbox"/>	
Geena	<input checked="" type="checkbox"/>	
Daniel	<input type="checkbox"/>	
Ben	<input type="checkbox"/>	
Ivy	<input type="checkbox"/>	
Hannah	<input type="checkbox"/>	
Frederick	<input type="checkbox"/>	
Ali	<input type="checkbox"/>	
	<input type="checkbox"/>	
	<input type="checkbox"/>	

**2. After marking all of your selections, detach the candidates lists (left side of cards).**

**3. Shred the candidates lists.**

**4. Feed your voting slips into the scanner.**

**5. Take your receipts.** Receipts can be used to confirm that you voted by visiting [votingstudy.rice.edu](http://votingstudy.rice.edu).

**Figure A2.1.** Voting Instructions for PaV

## Appendix 2--Pret a Voter Voting System Study Materials

<b>PRESIDENT AND VICE PRESIDENT</b>		<b>Card Key: 7rJ94K-1</b>	
<b>President and Vice President</b> <i>Vote for One</i>		Card 1 of 8	
REP	Gordon Bearce Nathan Maclean	<input type="checkbox"/>	Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.
DEM	Vernon Stanley Albury Richard Rigby	<input type="checkbox"/>	
LIB	Janette Froman Chris Aponte	<input type="checkbox"/>	
<b>CONGRESSIONAL</b>			
<b>United States Senator</b> <i>Vote for One</i>			
REP	Cecile Cadieux	<input type="checkbox"/>	
DEM	Fern Brzezinski	<input type="checkbox"/>	
IND	Corey Dery	<input type="checkbox"/>	
<b>Representative in Congress, District 7</b> <i>Vote for One</i>			
REP	Pedro Brouse	<input type="checkbox"/>	
DEM	Robert Mettler	<input type="checkbox"/>	
<b>STATE</b>			
<b>Governor</b> <i>Vote for One</i>			
REP	Glen Travis Lozier	<input type="checkbox"/>	
DEM	Rick Stickles	<input type="checkbox"/>	
IND	Maurice Humble	<input type="checkbox"/>	
Card 1 of 8			
<b>Ballot Continues on Card 2</b>			

Figure A2.2. Card 1/8 of the PaV ballot

## Appendix 2--Pret a Voter Voting System Study Materials

**General Election Ballot**  
**Harris County, Texas**  
**November 8, 2016**

After polls close, you can check your votes online: [votingstudy.rice.edu](http://votingstudy.rice.edu). Your ballot verification code is **7rJ94K**.

<p style="text-align: center;">Vote Verification Code: 7rJ94K1 Card 1 of 8</p> <p>Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.</p> <div style="display: flex; flex-direction: column; align-items: center;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div></div> <p style="text-align: center;">Card 1 of 8</p>	<p style="text-align: center;">Vote Verification Code: 7rJ94K2 Card 2 of 8</p> <p>Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.</p> <div style="display: flex; flex-direction: column; align-items: center;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div></div> <p style="text-align: center;">Card 2 of 8</p>	<p style="text-align: center;">Vote Verification Code: 7rJ94K3 Card 3 of 8</p> <p>Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.</p> <div style="display: flex; flex-direction: column; align-items: center;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div></div> <p style="text-align: center;">Card 3 of 8</p>	<p style="text-align: center;">Vote Verification Code: 7rJ94K4 Card 4 of 8</p> <p>Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.</p> <div style="display: flex; flex-direction: column; align-items: center;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div></div> <p style="text-align: center;">Card 4 of 8</p>
<p style="text-align: center;">Vote Verification Code: 7rJ94K5 Card 5 of 8</p> <p>Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.</p> <div style="display: flex; flex-direction: column; align-items: center;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div></div> <p style="text-align: center;">Card 5 of 8</p>	<p style="text-align: center;">Vote Verification Code: 7rJ94K6 Card 6 of 8</p> <p>Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.</p> <div style="display: flex; flex-direction: column; align-items: center;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div></div> <p style="text-align: center;">Card 6 of 8</p>	<p style="text-align: center;">Vote Verification Code: 7rJ94K7 Card 7 of 8</p> <p>Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.</p> <div style="display: flex; flex-direction: column; align-items: center;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div> <p style="text-align: center;">Card 7 of 8</p>	<p style="text-align: center;">Vote Verification Code: 7rJ94K8 Card 8 of 8</p> <p>Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.</p> <div style="display: flex; flex-direction: column; align-items: center;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div><div style="margin-top: 20px;"><div style="display: flex; align-items: center;"><input checked="" type="checkbox"/><div style="width: 100px; height: 15px; border: 1px solid black; margin-left: 5px;"></div></div></div></div> <p style="text-align: center;">Card 8 of 8</p>

Figure A2.3. PaV voter receipt

## Appendix 2--Pret a Voter Voting System Study Materials

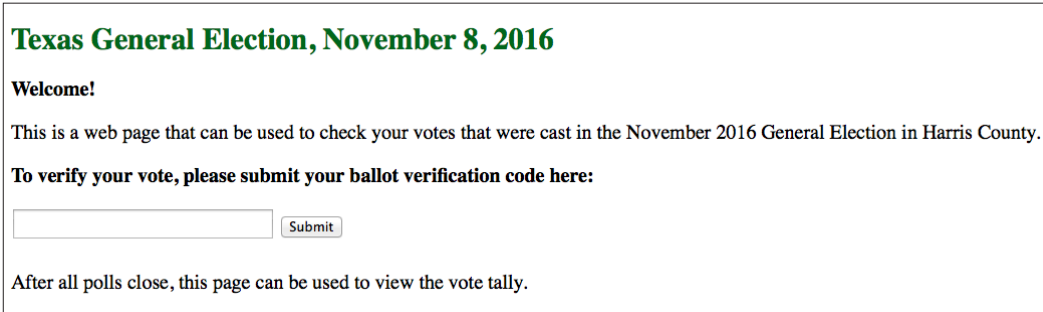


Figure A2.4. Screenshot of PaV's vote verification web page (site homepage)

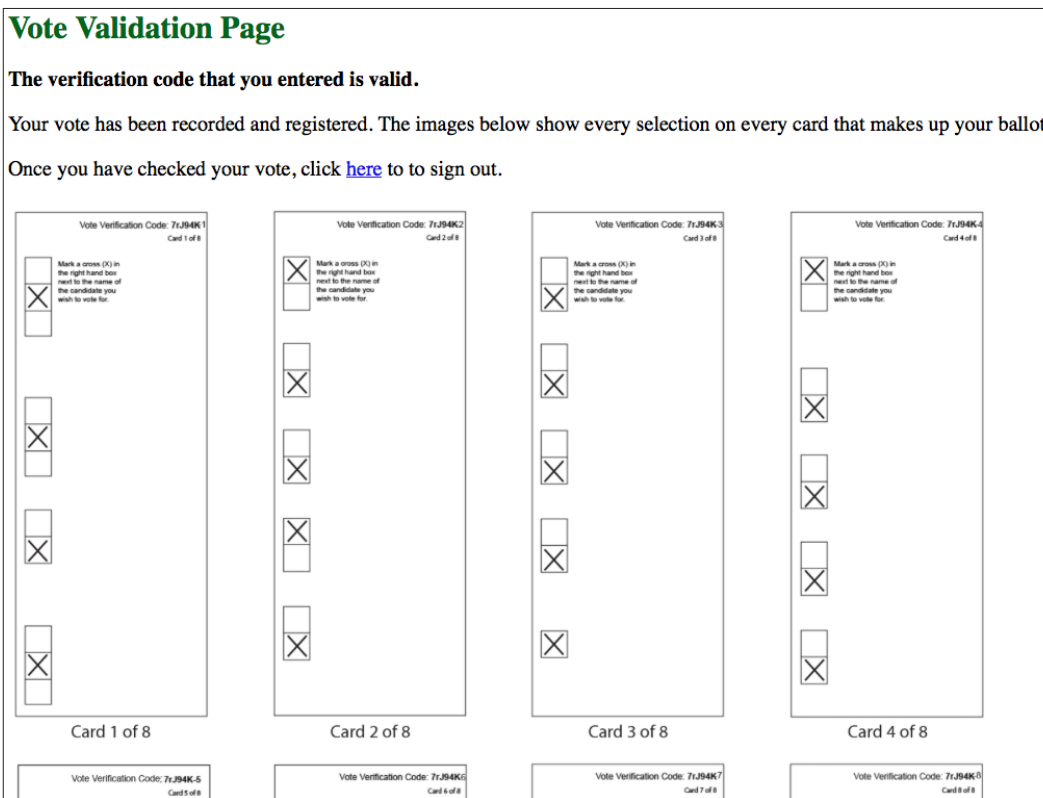


Figure A2.4. Screenshot of PaV's vote validation web page

### Appendix 3--Scantegrity II Voting System Study Materials

GENERAL ELECTION BALLOT HARRIS COUNTY, TEXAS NOVEMBER 8, 2016		
<p><b>- TO VOTE, COMPLETELY FILL IN THE OVAL <input type="radio"/> NEXT TO YOUR CHOICE.</b></p> <p>- Use only the special marking device provided.</p> <p>- If you make a mistake, do not hesitate to ask for a new ballot. If you make other marks, your vote may not count.</p> <p>- A confirmation number will appear inside the oval you mark. You may later use this confirmation number to verify your vote online. After marking the ballot, you may choose to write down your confirmation numbers on the card provided in the voting booth.</p> <p>- To cast your vote, take your ballot to the scanner. Keep the card to verify your vote online after the polls close.</p>		
PRESIDENT AND VICE PRESIDENT	STATE	COUNTY
<b>PRESIDENT AND VICE PRESIDENT</b> (Vote for One)  <input type="radio"/> Gordon Bearce REP <input type="radio"/> Nathan Maclean REP <input type="radio"/> Vernon Stanley Albury DEM <input type="radio"/> Richard Rigby DEM <input type="radio"/> Janette Froman LIB <input type="radio"/> Chris Aponte LIB	<b>COMMISSIONER OF GENERAL LAND OFFICE</b> (Vote for One)  <input type="radio"/> Sam Saddler REP <input type="radio"/> Elise Elzey DEM <b>COMMISSIONER OF AGRICULTURE</b> (Vote for One)  <input type="radio"/> Polly Rylander REP <input type="radio"/> Roberto Aron DEM	<b>DISTRICT ATTORNEY</b> (Vote for One)  <input type="radio"/> Corey Behnke REP <input type="radio"/> Jennifer A. Lundeed DEM <b>COUNTY TREASURER</b> (Vote for One)  <input type="radio"/> Dean Caffee REP <input type="radio"/> Gordon Kallas DEM
CONGRESSIONAL		
<b>UNITED STATES SENATOR</b> (Vote for One)  <input type="radio"/> Cecile Cadieux REP <input type="radio"/> Fern Brzezinski DEM <input type="radio"/> Corey Dery IND	<b>RAILROAD COMMISSIONER</b> (Vote for One)  <input type="radio"/> Jillian Balas REP <input type="radio"/> Zachary Minick DEM	<b>SHERIFF</b> (Vote for One)  <input type="radio"/> Stanley Saari GP <input type="radio"/> Jason Valle LIB
<b>REPRESENTATIVE IN CONGRESS</b> (Vote for One)  <input type="radio"/> Pedro Brouse REP <input type="radio"/> Robert Mettler DEM	<b>STATE SENATOR</b> (Vote for One)  <input type="radio"/> Ricardo Nigro REP <input type="radio"/> Wesley Steven Millette DEM	<b>COUNTY TAX ASSESSOR</b> (Vote for One)  <input type="radio"/> Howard Grady IND <input type="radio"/> Randy H. Clemons CON
STATE		NONPARTISAN
<b>GOVERNOR</b> (Vote for One)  <input type="radio"/> Glen Travis Lozier REP <input type="radio"/> Rick Stickles DEM <input type="radio"/> Maurice Humble IND	<b>STATE REPRESENTATIVE DISTRICT 134</b> (Vote for One)  <input type="radio"/> Petra Bencomo REP <input type="radio"/> Susanne Rael DEM	<b>JUSTICE OF THE PEACE</b> (Vote for One)  <input type="radio"/> Deborah Kamps <input type="radio"/> Clyde Gayton Jr.
<b>LIEUTENANT GOVERNOR</b> (Vote for One)  <input type="radio"/> Shane Terrio REP <input type="radio"/> Cassie Principe DEM	<b>MEMBER STATE BOARD OF EDUCATION DISTRICT 2</b> (Vote for One)  <input type="radio"/> Peter Varga REP <input type="radio"/> Mark Barber DEM	<b>COUNTY JUDGE</b> (Vote for One)  <input type="radio"/> Dan Atchley <input type="radio"/> Lewis Shine
<b>ATTORNEY GENERAL</b> (Vote for One)  <input type="radio"/> Tim Speight REP <input type="radio"/> Rick Organ DEM	<b>PRESIDING JUDGE TEXAS SUPREME COURT PLACE 3</b> (Vote for One)  <input type="radio"/> Tim Grasty DEM	<b>PROPOSITIONS</b> PROPOSITION 1 Without raising taxes and in order to pay for public safety, public works, parks and recreation, health care, libraries, and other essential services, shall Harris County and the City of Houston be authorized to retain and spend all city and county tax revenues in excess of the constitutional limitation on total city and county fiscal year spending for ten fiscal years beginning with the 2011 fiscal year, and to retain and spend an amount of city and tax revenues in excess of such limitation for the 2020 fiscal year and for each succeeding fiscal year up to the excess city and county revenue cap, as defined by this measure?  <input type="radio"/> YES <input type="radio"/> NO
<b>COMPTROLLER OF PUBLIC ACCOUNTS</b> (Vote for One)  <input type="radio"/> Therese Gustin IND <input type="radio"/> Greg Converse DEM	<b>PRESIDING JUDGE COURT OF CRIMINAL APPEALS, PLACE 2</b> (Vote for One)  <input type="radio"/> Dan Plouffe REP <input type="radio"/> Derrick Melgar DEM	

VOTE BOTH SIDES OF BALLOT

Ballot ID / Online Verification Number  
HC-2016-11-08-420795502

Figure A3.1. Scantegrity II ballot

### Appendix 3--Scantegrity II Voting System Study Materials

STATE	COUNTY
COMMISSIONER OF GENERAL LAND OFFICE (Vote for One)	DISTRICT ATTORNEY (Vote for One)
<input checked="" type="radio"/> <b>713</b> Sam Saddler REP	<input type="radio"/> Corey Behnke REP
<input type="radio"/> Elise Ellzey DEM	<input checked="" type="radio"/> <b>XRC</b> Jennifer A. Lundeed DEM
COMMISSIONER OF AGRICULTURE (Vote for One)	COUNTY TREASURER (Vote for One)
<input type="radio"/> Polly Rylander REP	<input type="radio"/> Dean Caffee REP
<input checked="" type="radio"/> <b>SIL</b> Roberto Aron DEM	<input checked="" type="radio"/> <b>SE6</b> Gordon Kallas DEM
RAILROAD COMMISSIONER (Vote for One)	SHERIFF (Vote for One)
<input type="radio"/> Jillian Balas REP	<input type="radio"/> Stanley Saari GP
<input checked="" type="radio"/> <b>222</b> Zachary Minick DEM	<input checked="" type="radio"/> <b>1EK</b> Jason Valle LIB
STATE SENATOR (Vote for One)	COUNTY TAX ASSESSOR (Vote for One)
<input type="radio"/> Ricardo Nigro REP	<input type="radio"/> Howard Grady IND
<input checked="" type="radio"/> <b>FG3</b> Wesley Steven Millette DEM	<input checked="" type="radio"/> <b>A31</b> Randy H. Clemons CON
STATE REPRESENTATIVE DISTRICT 134 (Vote for One)	NONPARTISAN
<input type="radio"/> Petra Bencomo REP	JUSTICE OF THE PEACE (Vote for One)
<input checked="" type="radio"/> <b>H11</b> Susanne Rael DEM	<input type="radio"/> Deborah Kamps
MEMBER STATE BOARD OF EDUCATION DISTRICT 2 (Vote for One)	<input checked="" type="radio"/> <b>T7H</b> Clyde Gayton Jr.
<input type="radio"/> Peter Varga REP	COUNTY JUDGE (Vote for One)
<input checked="" type="radio"/> <b>JH5</b> Mark Barber DEM	<input type="radio"/> Dan Atchley
	<input checked="" type="radio"/> <b>6C1</b> Lewis Shine
	PROPOSITIONS
	PROPOSITION 1

Figure A3.2. Photograph of a completed Scantegrity II ballot, with invisible ink confirmation codes revealed



## Appendix 3--Scantegrity II Voting System Study Materials

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

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 process.**

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 oval you mark.
2. **BEFORE YOU CAST YOUR BALLOT** record the Online Verification Number and the confirmation numbers below, using the special pen.

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

--

Race	Code
President And Vice President	
United States Senator	
Representative in Congress	
Governor	
Lieutenant Governor	
Attorney General	
Comptroller of Public Accounts	
Commissioner of General Land Office	
Commissioner of Agriculture	
Railroad Commissioner	
State Senator	
State Representative District 134	
Member State Board of Education, District 2	

Race	Code
Judge Texas Supreme Court	
Judge Court of Criminal Appeals	
District Attorney	
County Treasurer	
Sheriff	
County Tax Assessor	
Justice of the Peace	
County Judge	
Proposition 1	
Proposition 2	
Proposition 3	
Proposition 4	
Proposition 5	
Proposition 6	

3. **Cast your ballot as usual using the polling station's 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 County's vote verification web page: [mockelection.rice.edu](http://mockelection.rice.edu). 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 how you voted.

**Figure A3.3.** Scantegrity II vote verification sheet

### Appendix 3--Scantegrity II Voting System Study Materials

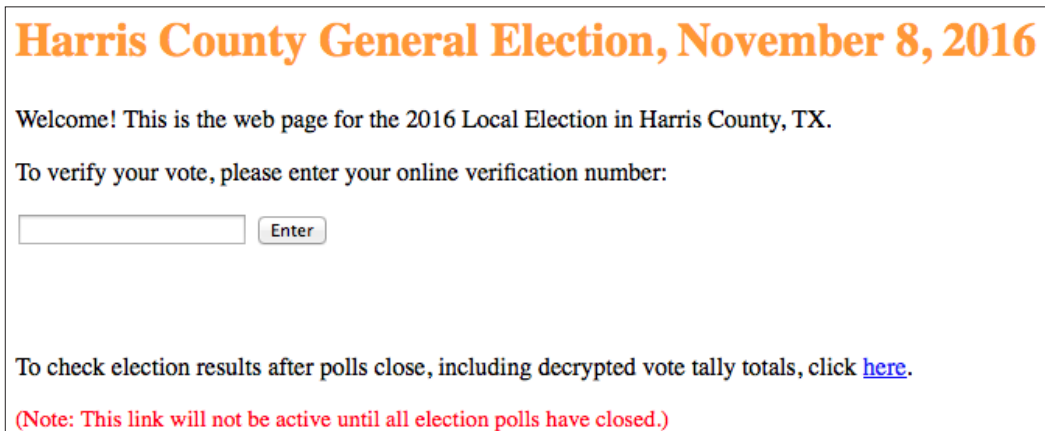


Figure A3.4. Screenshot of Scantegrity II vote verification page (site homepage)

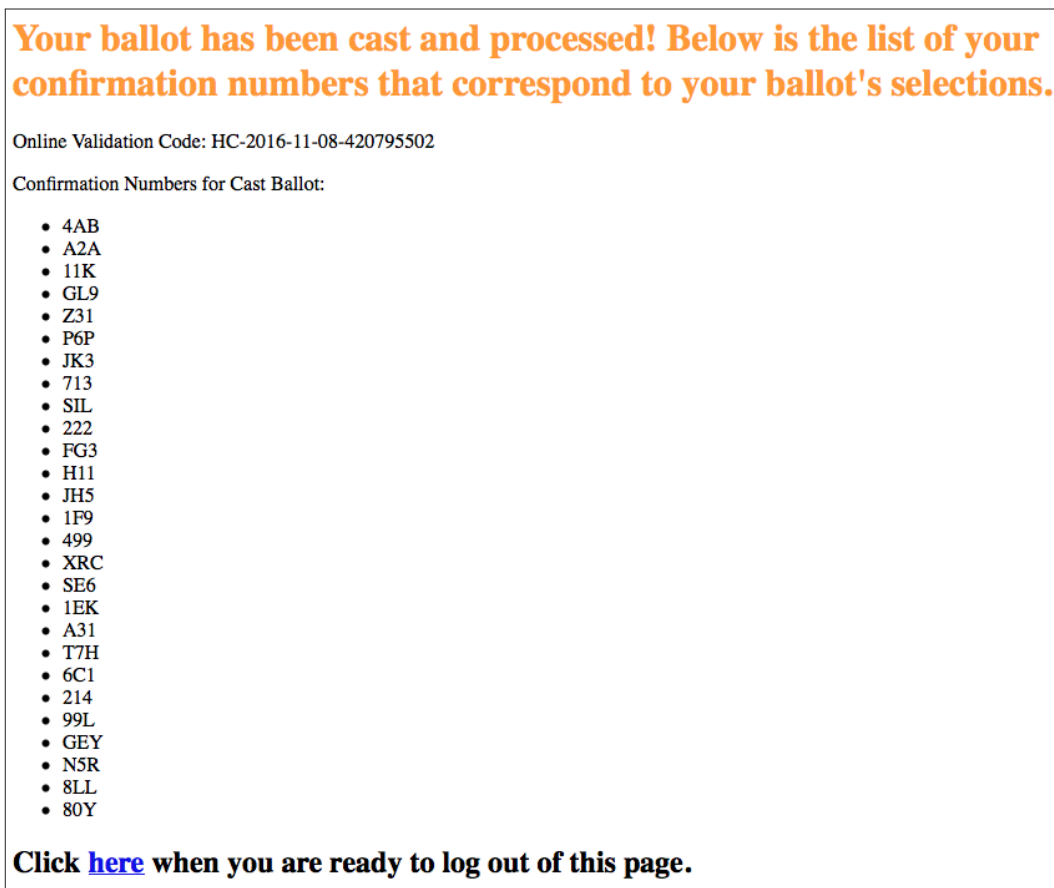


Figure A3.5. Screenshot of Scantegrity II cast ballot confirmation page

## Mitigating Coercion, Maximizing Confidence in Postal Elections

JACOB QUINN SHENKER, California Institute of Technology

R. MICHAEL ALVAREZ, California Institute of Technology

### 1. INTRODUCTION

Elections have traditionally depended on procedural safeguards and best practices to ensure integrity and instill trust. By making it difficult for individuals to manipulate ballots undetected, these policies electoral malfeasance. Even so, it is clearly preferable to move beyond this kind of best-effort security and instead provide strong guarantees of integrity and privacy.

An emerging literature on voting systems has identified two distinct approaches towards this end: build trustworthiness into the voting system, or audit the election after-the-fact to verify its integrity. The first strategy is embodied by end-to-end verifiable voting systems, which use cryptography to prove to the voter that their ballot was cast and tallied as intended (Chaum, 2004; Chaum, Ryan, & Schneider, 2005; Ryan, 2005; Adida & Rivest, 2006; Rivest, 2006). However, these systems are predicated on strong assumptions and use complicated, difficult-to-understand cryptography to deliver their security guarantees. Instead of attempting to provide these strict assurances, the auditing approach aims to output statistical evidence that an election was conducted properly (Stark, 2008, 2009; Aslam, Popa, & Rivest, 2007; Rivest & Shen, 2012).

Neither the literature on verifiable voting systems nor the one on post-election audits adequately addresses the problems specific to postal voting.<sup>1</sup> Indeed, the nature of postal voting makes an audit difficult. Any audit begins with a complete paper trail; in a postal election, where ballots can (and are) lost in the mail, it may be impossible to maintain a complete chain-of-custody regarding the postal ballots (Stewart, 2010). An audit can check the tally of ballots that were received, but this does not address postal voters' primary worry: that their ballots are being lost or tampered with in the mail.<sup>2</sup>

Since a key feature of end-to-end systems is that a voter may ascertain for themselves that their ballot was received unmodified, end-to-end verifiability should be a natural application to vote-by-mail. Yet previous work on end-to-end voting largely neglects voting by mail. This is lack of attention arises partly due to the difficulty of handling coercion in the postal voting.

Like any other remote voting protocol, postal voting allows much more pervasive coercion than is possible with in-person balloting. Researchers have designed many Internet-based end-to-end remote voting systems with coercion-mitigation techniques. In all of these systems, the voter is interacting with the system through their computer, which is capable of performing sophisticated

<sup>1</sup>We are not aware of any work on end-to-end auditing in postal voting (where nondelivery of ballots is detected by the audit.) We are aware of three voting system designs which apply cryptographic techniques to postal voting, but none address coercion: Popoveniuc and Lundin (2007) describe modifications to Punchscan and Prt Voter to make them suitable for use in postal elections; the Remotegrity (Zagrski et al., 2013) extension to Scantegrity II primarily targets electronic ballot return, but in principle could be used for mail-in Scantegrity ballots. Neither proposal is particularly attractive from a usability perspective: in the example one-race election given in the Remotegrity paper, the voter must use no fewer than six distinct authentication codes to cast and verify their ballot. A third proposal which looks promising, (Benaloh, Ryan, & Teague, 2013), is not strictly end-to-end verifiable but attains a high degree of verifiability with minimal cryptography, much in the spirit in the current work. It has been brought to the authors' attentions that Andrew Neff has commercialized a privacy-preserving postal ballot tracking product (Dategrity Corp., 2005).

<sup>2</sup>A survey of California postal voters indicates that many postal voters doubt that their ballots were delivered to the election authority: Bergman (2012) finds a full 18% of postal voters in California reported being either a little or not confident that their ballot was delivered safely, whereas 19% reported the same levels of confidence that their ballot was accurately counted and processed. Bergman notes that these two questions may measure the same underlying dimension, as there is a correlation  $r = 0.8$  and a Cronbach's alpha of  $> 0.7$ . This suggests that voters' doubts about their ballot counting can be largely explained by doubts about ballot transport. Other surveys bolster this claim, finding postal voters have lower confidence in elections across-the-board compared to in-person voters (Alvarez, Ansolabehere, et al., 2009).

cryptography. These systems leverage this ability to provide coercion-resistant voting; a paper-based protocol, as is needed for vote-by-mail, has no such recourse to sophisticated cryptography, since all cryptographic operations must be performed by the voter without computational aids. As such, vote-by-mail shares the main difficulty of Internet voting but cannot use the same mitigation techniques.

In this paper, we make a first attempt to consider the problem of coercion in the postal voting setting. We demonstrate that the defining features of postal voting constrain the design of any postal voting protocol, and thus many established techniques for end-to-end voting simply cannot be used (section 2). Along the way, we propose a scheme for providing auditability to vote-by-mail (section 3.3). While our resulting system does not provide *coercion-resistance* as defined by Juels, Catalano, and Jakobsson (2005), it provides a seemingly-weaker property of *coercion-evidence* of Grewal et al. (2013) (section 3.6). We argue that far from being weaker, this second property is more valuable in practice for convincing the electorate of the fairness of an election (sections 3.5 and 3.6).

Our design builds upon previous techniques. Our contribution is to recognize that the protocol of Grewal et al. (2013) works even if the ballot encryption step is postponed until after vote casting. This allows our system to offer two novel features: vote casting without cryptography and privacy-preserving publication of plaintext ballots.

The protocol we describe is not fully verifiable. However, given the increasing importance of securing vote-by-mail elections and the weaknesses inherent in traditional postal voting systems, we think it is an important step to bring vote-by-mail. To our knowledge, it is the first proposal to do this in a way that specifically addresses the problem of coercion, which otherwise would be a significant deterrent to its implementation in real-world elections. It remains to be seen in future work whether the techniques we describe can be profitably incorporated into a fully end-to-end verifiable voting scheme for postal voting.

Whether or not our particular design is worthwhile, it is undeniable that the postal voting problem has received disproportionately little attention, given its importance. In 1984, 4.5 million people voted by mail; in 2012, 21 million postal ballots were cast. In the intervening time, two states (Oregon and Washington) began conducting elections entirely by mail (Stewart, 2010). In November 2013, Colorado also began delivering postal ballots to all voters (Bland, 2014). Attracted by the promise of convenient voting, the electorate in these states strongly approves of vote-by-mail (Southwell, 2004; Alvarez, Ansolabehere, et al., 2009). Election administrators around the country are pushing for widespread implementation of mail-only elections as a way to curtail costs and increase turnout (Bergman, 2012; Gronke, Galanes-Rosenbaum, Miller, & Toffey, 2008). These efforts are bearing fruit: seventeen states already allow mail-only elections under special circumstances (NCSL, 2013).

This rapid growth will only exacerbate problems which are already being caused by postal voting. Many studies have shown that postal voting is less reliable than other methods by a number of metrics (Stewart, 2010; Alvarez, Stewart III, & Beckett, 2013) and that voters tend to trust it less than voting in-person (Bergman, 2012; Alvarez, Hall, & Llewellyn, 2008; Alvarez, Ansolabehere, et al., 2009).

Since passage of the Help America Vote Act, many California precincts have successfully improved their election infrastructure by replacing antiquated lever and punchcard machines by optical scanners, but these gains have been neutralized by a concurrent rise in no-excuse absentee and other forms of voting by mail; the rise of postal voting in California between 2000 and 2008 led to an additional 73,868 residual votes (Alvarez, Stewart III, & Beckett, 2013). Stewart (2010) estimates that in the 2008 election up to 3.9 million attempts to vote by mail did not result in a counted ballot; the resulting lost vote rate of 22% is more than five times the estimated overall lost vote rate. Given these statistics, the continuing and swift adoption of vote-by-mail poses important problems for election administration.

## **2. SYSTEM DESIGN**

### **2.1. The Power of Plaintext**

Any end-to-end voting system must prove to each voter that their ballot reached the ballot box unmodified. This proof must contain some information about how the voter has marked his ballot, otherwise the voter would have no assurance that the ballot received by the system was not altered. If this confirmation consisted simply of the voter's ballot choices, this would surely be adequate proof, but the voter could show it to a vote-buyer and use it to demonstrate complicity in a vote-selling arrangement. Proposals for electronic voting systems use cryptography during vote casting to get around this problem (e.g., designated verifier proofs), yet we are considering a paper-based protocol where these solutions do not apply (Jakobsson, Sako, & Impagliazzo, 1996; Hirt & Sako, 2000; Saeednia, Kremer, & Markowitch, 2004).

There is a solution suggested by JCJ and similar systems (Juels et al., 2005; Clarkson, Chong, & Myers, 2008; Bursuc, Grewal, & Ryan, 2012). During registration, the voter receives one true and a number of fake credentials. Whenever a voter submits a ballot, they include one of these credentials. All ballots are posted unencrypted, and this system may still be coercion-resistant provided we deny the coercer knowledge of which ballots have real credentials (which will affect the tally) and those which have fake credentials (which will not affect the tally).

In this way, we may reveal plaintext (unencrypted) ballots without compromising ballot secrecy nor coercion-resistance: a coerced voter may capitulate to the coercer's demand and vote according to their orders, but using a fake credential. They may then vote normally at another time using their true credential. As long as the adversary does not come to learn which credentials are true and which are fake, he cannot distinguish compliance from noncompliance. (We will argue in the next section that the requirements of the registration phase force us to abandon the distinction between true and fake credentials, and make modifications accordingly. But for now we consider a system that does have distinct true/fake credentials.)

Of course, verifiability requires that the system prove to the voter that their true vote was counted while their fake votes were not. By posting plaintext ballots, the system may demonstrate that ballots reached the ballot box unmodified without using cryptography; the system then uses cryptography to prove that only the true ballots in the ballot box were tallied. In the case where only encrypted ballots are posted, cryptographic proofs are needed for both steps. In a strict sense, showing the voter his plaintext ballot is no better than publishing encrypted ballots, since cryptography is required for full verifiability either way. However chief among the doubts of a postal voter, unlike a polling-place voter, is ballot transport: will his ballot make it to the ballot box unmodified? Seeing a publicly-posted image of his ballot, just as he marked and submitted it, would give him substantial confidence that his ballot was received unmodified.

The ability to post plaintext ballots provides additional advantages. Consider that election authorities are free to publish both the ballot scan itself as well as how the ballot was interpreted by the optical scanner or canvassing board. This allows anyone, not just government-approved auditors, to examine disputed ballots for themselves. Previous experience with election auditing suggests this capability would do much to increase election trustworthiness. Election transparency advocates in Humboldt County, California, with the cooperation of the County Clerk, scanned and publicly posted anonymous ballot scans after two 2008 elections, discovering 197 lost votes that were missed by ballot tabulation software (Greenson, 2009). The ability of anyone to audit every aspect of the election up until the final tally, which requires distinguishing between true and fake credentials so as to only count ballots with true credentials (using cryptography to do this in a verifiable way is discussed in section 3.3).

Notice that using a credential system that allows for overriding or cancelling votes, e.g., JCJ's true/fake credentials, is the only way to enable the public auditing of ballot scans in a coercion-resistant way, because any scheme that involves posting full ballot scans is susceptible to a pattern voting attack or the inclusion of intentional identifying marks on the ballot. Using true/fake creden-

tials, we can allow an adversary to trace a ballot back to a particular voter, because the adversary still will not know if that ballot was submitted with a true or fake credential.

So far we have assumed that we may communicate true and fake credentials to each voter without an adversary learning them. To do this, we must register voters over an *untappable* communication channel. Almost all previous voting systems have assumed the existence of such a channel, but recently it has become clear that technology has made this assumption more dubious, and that channels once thought to be effectively untappable in practice can in fact be tapped en-masse and at low cost (Benaloh, 2013; Nixon, 2013). In the next section, we discuss modifications to the true/fake credential scheme that allows it to mitigate coercion even without using an untappable channel. In particular, we find that in the absence of an untappable channel the system cannot make a distinction between true and fake credentials, so all issued credentials must be equally valid. Instead of true votes overriding fake votes, we adopt a scheme whereby any credential may cancel the vote of any other credential issued to the same voter.

## 2.2. Registration Phase

Registration is the process by which voters authenticate themselves to election authorities before an election. In the US, this usually involves the voter providing their address, social security number or driver's license number, and their signature. When it is time to submit a ballot, reproducing these personal data on their ballot serves as a voter's proof to the election authority that they are authorized to cast a vote.

The registration phase takes on special importance in the context of an end-to-end verifiable, coercion-resistant voting system because both the end-to-end verifiability and coercion-resistance mechanisms depend on the election authority and the voter sharing cryptographic secrets before the election. The verifiability and privacy guarantees of these systems are predicated on the perfect untappability of the communication channel between the voter and the registrar, and if that assumption is broken so too are those guarantees. That is to say, such a voting system is only as secure as its registration phase. Almost all previous work on end-to-end voting systems assumes that voters are able to register via mail or in-person in a perfectly secure way.

However, a perfectly secure channel is not necessary if intrusions or delivery failures can be detected and mitigated. Consider the election registrar who wishes to securely transmit a voting credential to each voter. If the registrar sends the credential in a tamper-proof sealed envelope, an adversary may intercept the credential mailing, but it will either be delivered with a broken seal or will not be delivered at all. In either case, the intrusion would be detected.

Since an adversary may intercept any given credential mailing and prevent the voter from receiving it, we assume that the registrar sends enough credentials so that it is highly likely that the voter receives at least one.<sup>3</sup> In doing so, the registrar has transmitted at least one credential to the voter that was not intercepted by an adversary. One might think that our task, of securely communicating a credential to the voter, is thus accomplished. However, in the process the adversary may have intercepted a number of credentials. One might respond that the voter could notify the registrar which credential mailing succeeded in reaching him unintercepted, but this is not possible, because an adversary in possession of a valid credential is indistinguishable from a voter in possession of a valid credential. The only way out would be to presuppose some secret information shared by the voter and registrar that the voter may use to authenticate himself, which is merely begging the question.

We have thus found a way to communicate a credential securely to a voter, but a number of equally-valid credentials may be intercepted by an adversary. To the election authority, these credentials are indistinguishable, so they may all be used to cast a ballot. Regrettably, this means an adversary may cast a ballot on behalf of a valid voter. Note, however, that we would expect most registered voters to cast ballots. If a voter was observed to have voted once with one credential, and again with another credential, we may suppose that one of those credentials was intercepted by an adversary since there would be no legitimate reason for a single voter to cast multiple ballots.

---

<sup>3</sup>In addition, one can allow voters to request additional credentials.

This observation inspired the notion of *coercion-evidence*, introduced by Grewal et al. (2013). In our implementation, the registrar assigns to each voter a set of credentials. If more than one ballot is submitted using credentials in the same credential set, these ballots are not counted; instead, they are set aside and marked as evidence of coercion. The system publicly outputs the tally (not including cancelled votes) as well as the number of cancelled votes without disclosing which or whose ballots have been cancelled.<sup>4</sup> In the following sections, we discuss how to use cryptographic techniques to do this in a verifiable way (section 3.3), how these cancelled votes can be used in a post-election audit (section 3.5), and how this approach to coercion mitigation compares with the more common notion of coercion-resistance (section 3.6).

### 3. SYSTEM DESCRIPTION

#### 3.1. Preliminaries

*credential*. A human-readable password that a voter includes with their ballot in lieu of their name, signature, or any other identifying information.

*bulletin board  $\mathcal{BB}$* . An append-only bulletin board listing public election data, presumably hosted on the election authority's website.

*threshold encryption scheme*. An encryption scheme wherein the secret key is distributed amongst a set of trustees, wherein a threshold fraction (e.g., a majority) of trustees need to cooperate to decrypt a ciphertext (Brandt, 2006).

*plaintext equivalence test (PET)*. Given two ciphertexts encrypted with the same public key, a multiparty protocol may be performed by the trustees to prove whether their corresponding plaintexts are equal without revealing the decryption key or the underlying plaintexts. (Jakobsson & Juels, 2000).

*verifiable reencryption mixnet*. A mixnet which takes as input a list of ciphertexts and outputs a permuted list of reencryptions of those same ciphertexts; it outputs transcripts that are sufficient to verify that the shuffle has been performed correctly. (Chaum, 1981; Jakobsson, Juels, & Rivest, 2002).

*trustees*. A set of entities that execute a series of distributed protocols to process the election data.

*registrar*. A trusted entity responsible for maintaining the voter rolls and putting tamper-evident seals on credential mailings.

We adopt a randomized threshold encryption scheme with a plaintext equivalence test, such as distributed El Gamal (Elgamal, 1985; Brandt, 2006). We write  $\{\text{plaintext}\}_{PK}^r$  to mean the ciphertext produced by encrypting *plaintext* with the public key *PK* and randomness *r*.

#### 3.2. Assumptions

We make the following assumptions:

1. At majority of trustees are honest. A majority of trustees may generate arbitrarily malformed credential data, including additional credentials for vote-stuffing, or may decrypt any encrypted data. In particular, they may track ballots through the mixnet, and thus discover which ballots were cancelled.
2. The registrar is honest. This is not nearly as strong an assumption as it seems: in any voting system, there must be some entity which decides who is allowed to vote and maintains the voter rolls. Only the registrar knows the real-world voter identities (i.e., names and addresses).
3. The envelopes in which credentials are mailed satisfy two properties: a voter may ascertain that an envelope has not been opened, and that the provenance of the envelope can be perfectly authenticated. In this way, an adversary cannot intercept the contents of the envelope without being

<sup>4</sup>Again, implementation of this system would allow a process for voters to request additional credentials, as well as a process whereby the voter can identify herself in person to the election authority and cast a final ballot that could be included in the tally were all of the ballots associated with her previously-issued credentials used by coercers. These procedures would ensure that coerced voters do not lose their ability to vote.

detected. The former may be attained using a tamper-evident seal; the latter may be satisfied using any techniques for authenticating paper documents (e.g., the security features used in paper money).

4. Malware cannot spoof the bulletin board; standard Internet security techniques may be used to prevent this possibility.
5. Ballots are divisible into sections ( $B_k^l$  in the notation introduced below) such that no ballot is uniquely identified by a voting pattern on any given one ballot section. It is left to future work to adapt the protocol to handle write-in candidates or rich ballot types such as IRV.

### 3.3. Protocol

1. Trustees participate in a distributed key generation protocol, publishing a public key  $PK$  to  $\mathcal{BB}$  in such a way as no minority of trustees can reconstruct the private key.
2. Trustees execute a multiparty oblivious printing protocol (Essex & Hengartner, 2012) to generate and print credential mailings in invisible ink.<sup>5</sup> This protocol outputs both cryptographic information (posted to  $\mathcal{BB}$ ) and physical credential mailings.
3. As part of the oblivious printing protocol, trustees generate a list of credentials,  $(\text{voter}_i, \{\text{cred}_{ij}\}_{PK}^{r_{ij}})$ , where  $\text{cred}_{ij}$  is the  $j$ -th credential associated to voter  $i$ .
4. The oblivious printing protocol also outputs credential mailings, where the  $j$ -th credential mailing for a voter  $i$  includes the plaintext credential  $\text{cred}_{ij}$  printed in invisible ink on both an adhesive label and a receipt slip, both enclosed in a tamper-evident sealed envelope.
5. The registrar is assumed to begin with a list of voter ID numbers and their mailing addresses,  $(\text{voter}_i, \text{address}_i)$ .
6. For each printed credential mailing for voter  $i$ , the registrar fingerprints (using, e.g., Sharma, Subramanian, and Brewer (2011)) a blank sheet of paper, delivers it to the first trustee to be printed and requests a credential for voter  $i$ ; when it is returned by the last trustee, with the credential fully printed, the registrar verifies that the invisible ink has not been activated and that the returned sheet was the same one it delivered (using the fingerprint). This prevents the last trustee from revealing the invisible ink, copying down the credential, and printing an identical copy to mail off; this would allow silent interception of credentials, which the protocol must prevent.
7. The registrar then seals the credential mailing in an envelope with a tamper-evident seal and mails it to  $\text{address}_i$ .
8. Before the election, the trustees generate and print a large number of credentials, and the registrar mails each voter one of these credentials. The registrar sends each voter additional credentials from this set periodically during the election period, or at the request of the voter.
9. Trustees post on  $\mathcal{BB}$  a commitment to  $(\{\text{cred}_{ij}\}_{PK}^{r_{ij}}, \{\text{voter}_i\}_{PK}^{p_{ij}})$ , a list of encrypted credentials and the encryption of their associated voter identities.
10. During the balloting period, voters download a ballot form from  $\mathcal{BB}$ , print it, and fill it out. The voter then chooses any of the credential mailings they have received, opens it, and uses a special pen to activate the invisible ink on the mailing to reveal the credential. Verifying that the credential on the receipt slip matches the credential on the adhesive label, they place the label on the ballot and mail it back; they keep the receipt slip so that they may find their ballot on  $\mathcal{BB}$  when its scan is posted.
11. After the balloting period has closed, trustees open the commitment to  $(\{\text{cred}_{ij}\}_{PK}^{r_{ij}}, \{\text{voter}_i\}_{PK}^{p_{ij}})$ . Furthermore, trustees jointly decrypt each  $\{\text{cred}_{ij}\}_{PK}^{r_{ij}}$  and post proofs of correct decryption to  $\mathcal{BB}$ . By associating each decrypted plaintext credential with its encryption, the trustees then post

<sup>5</sup>The protocol is a generalization of the usual two-party visual cryptography scheme (Chaum, 2004), but extended to distribute trust amongst multiple printers. The printers each generate shares of a secret (in our case, a credential); each printer in turn prints its share in invisible ink, so that printers may not read previously-printed shares as they are printing their own. After all of the printers have printed their share, the invisible ink may be developed with a special pen to reveal the secret. The protocol guarantees the printing will be correct unless a majority of trustees conspire, and that none of the printers will know the secret (Essex & Hengartner, 2012).



- $(\text{cred}_{ij}, \{\text{voter}_i\}_{PK}^{p_{ij}})$ . Note that here it is crucially important that each voter identity  $\{\text{voter}_i\}_{PK}^{p_{ij}}$  is encrypted with unique randomness; otherwise, by matching plaintext credentials with the same voter identity ciphertext, credentials belonging to the same voter could be linked.
12. The election authority scans all ballots, posting each ballot's credential, image, and textual representation on  $\mathcal{BB}$ . We write  $B_k^l$  for the ballot data corresponding to the  $l$ -th race on the  $k$ -th ballot and  $\text{cred}_k$  for the credential included on the  $k$ -th ballot. Using the output of step 5, an encrypted voter identity  $\{\text{voter}_i\}_{PK}^{p_{ij}}$  may be associated to each ballot, where  $\text{cred}_{ij} = \text{cred}_k$ .
  13. For each race  $l$ :
    - i. Trustees post  $(\{\text{voter}_i\}_{PK}^{p_{ij}}, \{B_k^l\}_{PK})$  to  $\mathcal{BB}$ .
    - ii. Trustees execute a verifiable reencryption mixnet to shuffle  $(\{\text{voter}_i\}_{PK}, \{B_k^l\}_{PK})$ , posting the transcript and proofs to  $\mathcal{BB}$ .
    - iii. Trustees execute plaintext equivalence tests between the encrypted voter identity for each pair of ballots, posting transcripts to  $\mathcal{BB}$ . The result is a list  $(\{\text{voter}_i\}_{PK}, \{B_{n_1}^l\}_{PK}, \{B_{n_2}^l\}_{PK}, \dots)$  where  $B_{n_i}^l$  are the ballots whose encrypted voter identities have been shown to be plaintext equivalent.
    - iv. Trustees jointly decrypt ballot information  $B_k^l$  in the case where only one ballot has been associated with a given voter identity, and post it to  $\mathcal{BB}$  along with a proof of correct decryption. The tally is simply the sum of these.
    - v. The final output is the tally, the decrypted ballot information  $\{B_k^l\}$  for non-cancelled votes, and the number of cancelled votes (number of voter identities corresponding to cancelled ballots, *not* the number of cancelled ballots).

### 3.4. Attacks or Errors Prevented

- **Trustees adding or deanonymizing ballots.** No minority of the trustees can add a valid ballot to  $\mathcal{BB}$  (since doing so would require generating a new credential, which requires the cooperation of a majority of trustees). Similarly, no minority of the trustees can associate a ballot with a voter (since doing so would require). Note that a majority of trustees still can do so, and this ability may be desirable for the purpose of investigating coercion after the fact.
- **Malformed credential mailings.** The oblivious printing protocol includes verifiability steps to ensure that credential mailings are printed correctly unless a majority of trustees conspire (Essex & Hengartner, 2012). We assume that a voter can distinguish valid credential mailings sent by the election authorities from spoofing attempts sent by an adversary.<sup>6</sup>
- **Removing or modifying ballots.** Any voter can look up the ballots corresponding to their credentials and verify that they match the ballots they submitted. The voter may make scans or copies of their ballots before submitting them if they wish, and they may use these as evidence of manipulation in case  $\mathcal{BB}$  does not contain matching ballots.<sup>7</sup>
- **Deanonymization via bubble fingerprinting.** The coercion-mitigation property holds even if voters may voluntarily deanonymize their ballots, because voters will know to adhere to the vote-buyer's or coercer's demands in the deanonymized ballot but may submit a second, unidentifiable ballot to cancel it. It no longer holds if voters accidentally make their ballot identifiable, because the voter will not know to cancel their ballot. Calandrino, Clarkson, and Felten (2011) describe a machine-learning procedure that could be able to link ballots to individuals by examining the way in which they fill in the optical-scan bubbles. To combat this, ballot scans could be posted with the actual marks blurred or masked by solid black squares. To ensure that this masking is done correctly, a cut-and-choose-style protocol could be used: a limited number of bubbles could be unmasked, selected using a trusted random beacon, such as stock market data (cf. Clark, Essex,

<sup>6</sup>This can be done using well-known techniques, for example, those used in authenticating paper currency.

<sup>7</sup>Forensic techniques such as paper fingerprinting (Sharma et al., 2011) may be of use in proving that their ballot has been manipulated.

- and Adams (2007), Clark and Hengartner (2010)). The number of unmasked bubbles per ballot would be chosen so that they would provide insufficient data for a Calandrino-style attack.
- **Misprinted ballots.** Ballot images are posted on  $\mathcal{BB}$ , so anyone may verify that the ballot was printed correctly and that the ballot design complies with election law.
  - **Malicious optical scanner or canvassing board.** A textual representation of each ballot will be posted together with a high-resolution image of each ballot on  $\mathcal{BB}$ . Consistency between the two can be checked manually or with the assistance of ballot-auditing software (Kim et al., 2013). The textual representation is the output of the optical scanner, or in the case of a dispute, the interpretation of the canvassing board, and as such both may be verified. To our knowledge, this is only voting system which allows the publishing of ballot scans in a way unsusceptible to coercion, and as such is the only system that allows canvassing board decisions to be audited by anyone.
  - **Active coercion.** A voter can comply with any request the coercer makes, including pattern voting or abstention, and can turn over all of their credentials. The voter may then obtain a new credential and use it to submit another ballot, thus cancelling the coerced vote. The only way to successfully and undetectably coerce a voter is to intercept all of their communications from the beginning of the registration period (when the first credential is distributed) to the end of the balloting period; since this time period may be months or years, it would require enormous resources to coerce a significant number of votes.
  - **Vote selling.** Again, a voter can reveal to a vote-buyer all their credentials and all of their submitted ballots, and the vote-buyer can indeed verify that these ballots appear on  $\mathcal{BB}$ , but the vote seller may at any time obtain a new credential and use it to submit another ballot, thus cancelling the sold vote and marking it as coercion-evidence. Note that voter-sellers are disincentivized from allowing sold votes to count, since if they sold their vote once, they could sell it again to additional vote-buyers; the multiple ballots submitted by these vote-buyers will all be cancelled. Thus, the price of a sold vote will be driven to zero.
  - **Loss of privacy.** Because the registrar distributes credentials to the voter in a tamper-evident sealed envelope, a voter can trust that any credential mailing that arrives intact has not been intercepted. Thus, the only way an adversary can learn of a voter's true vote is if he discovers all of the voter's credential mailings after the voter has opened them. By hiding his credential mailings, a voter can make it arbitrarily difficult for his privacy to be violated. Note that the voter can always voluntarily give up privacy by revealing their credentials, but as we have seen above, this ability does not make them susceptible to coercion, since revealing credentials only reveals the ballots the voter has submitted using those credentials; there is no guarantee that any of those ballots would count.
  - **Forced abstention or retribution.** Forcing abstention or exacting retribution require the adversary to learn at least one credential with which the voter has submitted a ballot. We have seen above that a voter can make this arbitrarily difficult. Additionally, a voter strongly afraid of coercion or retribution may implement the following strategy: he may obtain a number of credentials, submit blank ballots for all of them, and immediately afterwards destroy the credential mailings. Without the cooperation of a majority of trustees, the only way the adversary can learn the credentials the voter used is by intercepting the blank ballot mailings themselves. Furthermore, as long as two of the blank ballots are not intercepted, they will be marked as coercion-evidence.
  - **Silent coercion.** A voter is said to be *silently coerced* if he is coerced without his knowledge (Grewal et al., 2013). A voter may be silently coerced if the adversary intercepts one of the voter's credentials and vote on his behalf without the voter's knowledge. These silently coerced votes will only count if the voter does not submit any ballots of their own. This makes our system, along with Caveat Coercitor (Grewal et al., 2013), one of the few systems that handle this kind of coercion.
  - **Information leakage.** The above attack mitigations and privacy guarantees are predicated on the assumption that an adversary cannot learn which ballots are cancelled and which are not. The full set of ballots, the tally with cancelled votes removed, and the number of cancelled votes are

public information; in contrived cases, this information is sufficient to determine which ballots were cancelled and which were not. In Appendix A, we discuss this vulnerability and provide a simple remedy.

### 3.5. Error Recovery

Our protocol was designed to allow voters to see if their ballot was received intact by looking for it on  $\mathcal{BB}$ . If voters self-report missing or modified ballots, this information is included in an audit trail. If a significant number of complaints are received, the election authority or independent auditors may be prompted to investigate further. However, voters' self-reports cannot be assumed to be perfectly trustworthy or reliable. At the cost of a more complex procedure, we can do better, by allowing voters to correct these errors (resubmit their ballots until they are properly received) instead of merely declare them.

To do this, the system can post partial credentials<sup>8</sup> of the ballots as they are received; voters can check that their ballot was received, and can submit another if necessary. Note that this could lead voters to unintentionally cancel their own vote. Because of delays in postal service, a voter could see that their first ballot is missing from  $\mathcal{BB}$  and proceed to submit a second one; if the first ballot is not lost, but merely delayed, the election authority will eventually receive both and cancel the vote. This can be prevented by instituting a policy of disqualifying ballots if they are received a certain amount of time (for example a week) after they were postmarked.<sup>9</sup> This way, a voter knows that he must resubmit a ballot if it does not appear on  $\mathcal{BB}$  within a week of submission, and can be sure that this resubmission will not unintentionally cancel his vote. This protocol guarantees voters the ability to reliably cast a ballot even in the face of inconsistent postal service.<sup>10</sup>

Coercion will lead to ballots being cancelled; we now argue that this cancellation procedure prevents coercion from manipulating the outcome of an election. Consider the four regimes jointly characterized by the level of actual coercion (high or low) and the number of cancelled votes (fewer than the margin of victory, or in excess of the margin of victory).

In the low-coercion few-cancelled-votes regime, the cancelled votes would be due to a handful of instances of actual coercion or simply a few voters mistakenly submitting more than one ballot. These few cancelled votes would change the published tally slightly from the tally of voters' true preferences, but only by a small number of votes relative to the margin of victory, so would not come close to changing the election outcome or significantly modifying the margin of victory.

We now consider the case in which there are many cancelled votes, comparable to or exceeding the margin of victory, but little actual coercion. This means that there are many ballots being cancelled for reasons other than coercion: voters could be submitting multiple ballots themselves, or they could be publicly revealing their credentials so that others may cancel their vote for them. Based on existing research, we do not believe that many voters will intentionally cancel their ballots.<sup>11</sup>

<sup>8</sup>A partial credential is a truncated credential, where enough of the credential is posted so that it is uniquely identifiable but an adversary cannot efficiently brute-force guess the full credential. If full credentials are posted during balloting, then anyone can submit a ballot with any of these credentials, cancelling a vote.

<sup>9</sup>Disqualified ballots are still posted, but are marked as such, and are neither tallied nor can they cancel votes. To detect if the system is adversarially disqualifying ballots by falsely claiming they were received after the one-week deadline, scans of their enclosing envelopes (with the date they were postmarked) can be posted along with the ballots. It will then be evident if there are an abnormal number of such ballots. Additionally, the system can post the scans of these envelopes on  $\mathcal{BB}$  before they are opened and the enclosed ballots scanned, so the system cannot preferentially disqualify ballots for a given candidate.

<sup>10</sup>Note that this has the undesirable feature of publishing a running tally of all ballots, including cancelled ballots, during the voting period. To prevent this, instead of posting partial credentials and ballot scans to  $\mathcal{BB}$  during the voting period, one could instead publish partial credentials, a cryptographic commitment to the ballot scan, and the ballot scan itself encrypted with the full credential as encryption key. In this case, only those in possession of the full credential (by arguments above, only the voter) may examine his plaintext ballot scan. After the election, the commitments to all of the ballots are opened and anyone can examine (and audit) any ballot image, preserving the auditability properties discussed below.

<sup>11</sup>Some nations, including Sweden and Estonia, have procedures that allow voters to cast multiple ballots, with later ballots overriding earlier ones. Estonia's revoting process is a close analogue to what we propose here (ENEC, 2013a). Importantly, data from recent elections in Estonia have shown very low levels of revoting; for example in the 2011 Estonian parliamentary

In the opposite regime, where there are few cancelled ballots but high levels of coercion, many instances of coercion are not being detected. Either voters are knowingly being coerced and are simply choosing not to submit a second ballot to cancel the coerced votes, or coercers are intercepting many credentials from non-voters and using those to cast votes on their behalf without the non-voter's knowledge.<sup>12</sup> The former does not seem likely, so we consider the latter. The worst case arises if coercers are able to use demographic information and voter profiles to selectively target potential non-voters for credential interception. Even so, unless they are able to do this selection with close to perfect accuracy, there will be some fraction of suspected non-voters who will end up submitting a ballot themselves. These ballots will show up as cancelled votes, since both the coercer and the voter have submitted ballots for the same voter identity, so as long as the coercer is submitting a significant number of ballots on behalf of potential non-voters, this will arise in an anomalously-high cancellation rate, signaling election authorities or independent auditors to investigate the reason for these cancelled votes. Note that in low-turnout situations the margin of victory may exceed the number of cancelled votes; however, the number of cancelled votes would still have to be high in absolute numbers. Thus, the anomaly would be detected and further investigation would be prompted.

Similarly, in the high-coercion, high-cancellation regime, the system would announce a large number of cancelled votes, and election authorities would be prompted to investigate the detected coercion. The adversary *does* succeed in casting doubt on the integrity of the election. In this sense, our vote cancellation procedure does not seem sufficient to hold a fair election in this situation. However, in the presence of a high cancellation rate due to widespread coercion or suspected government corruption, cryptography would do little to dispel a lay voter's distrust of the outcome (especially since the government was likely involved with designing and implementing the voting system in the first place). Instead of cryptographic assurances that may be of little real value in convincing the public of a correct outcome, our system is highly transparent: it outputs a variety of information which will be useful in identifying coerced ballots and ensuring a correct election outcome. Publicly-accessible ballot scans, the physical ballots themselves, and the number of cancelled votes (potential markers of coercion) comprise an extensive audit trail which may be used by auditors or in litigation addressing election impropriety. Furthermore, if it is desired, the protocol can allow auditors to deanonymize certain ballots. This would allow them to study ballots which have been cancelled and thus potentially coerced.<sup>13</sup>

In much the same spirit as a risk-limiting audit, a protocol may be agreed upon specifying how to determine an election outcome given this audit trail. In this way, elections under doubt would be handled in the courts, much the way they are now, but our system would provide direct information about coercion. A coerced voter could be assured that by submitting a second ballot to cancel his vote, he has announced his plight to the election authorities and they will follow this agreed-upon procedure for ensuring that this coercion does not manipulate the election outcome.

### 3.6. Beyond Coercion-Resistance

Coercion-resistant voting systems offer a mechanism which allows voters to pretend to acquiesce to a coercer's demand while actually voting how they please. The mathematical formulation of

---

elections, 4,384 multiple Internet votes were recorded, and only 82 Internet votes were cancelled by a later paper ballot (of a total of 140,846 Internet voters) (ENEC, 2013b). We have no reason to expect that there would be a greater incidence of revoting in our case, where revoting is not allowed (as it cancels the vote). Thus, attempts at intentional multiple voting in our system could be seen as protest voting, but again there is little evidence in the research literature that shows a great deal of protest voting in existing electoral systems, and we do not expect that protest voting would be more prevalent in our system. See Stiefbold (1965) for a classic discussion of protest voting and void ballots; or Sinclair and Alvarez (2004) for a more recent examination of intentional voiding of ballots.

<sup>12</sup>We call this *silent coercion*, following Grewal et al. (2013).

<sup>13</sup>There is precedent for this kind of deanonymization for the purpose of election forensics: in certain jurisdictions, such as the U.K., the government is legally obligated to deanonymize certain ballots at the request of an election judge (Smart & Ritter, 2009).

this property, introduced by Juels et al. (2005), states that in the course of the execution of such a protocol, an adversary is not able to learn any additional information beyond the tally itself. Put another way, the voting system does not allow the coercer to distinguish between a coerced voter's compliance and non-compliance. As such, the voter need not heed the coercer's threat, and may vote according to their true preferences. The precision of this property is appealing; it purports to perfectly mitigate the threat of voter coercion. However, in practice a coercion-resistant voting system could fall significantly short of this goal.

Any coercion-resistant voting system presupposes an untappable channel, yet none of the channels over which remote elections are conducted—mail, phone, and the Internet—are perfectly untappable.<sup>14</sup> As recent descriptions of state-sponsored surveillance programs have illustrated, long-term, mass interception of mail is not a theoretical threat (Nixon, 2013). One might respond that instead of registering remotely, we could mandate in-person registration, which is surely more secure.

Benaloh (2013) argues that even this is not good enough, observing that the prevalence of cell phones and wearable cameras prevents even a polling booth from being truly private. Given that coercers or vote-buyers can instruct voters to surreptitiously record their registration or balloting sessions with these cameras, even the paradigmatic untappable channel (the private voting booth) is no more. Without any untappable channels, perfect coercion-resistance is impossible. Benaloh concludes that surrender is not an attractive option, but there seems to be little point to adding significant complexity to election protocols in an increasingly futile attempt to defeat pre-election coercion.

Instead of surrender, we advocate a strategic retreat. Our vote-cancellation procedure will still detect coercion perfectly even without an untappable channel. Once detected, an audit (and associated litigation) can use this information to neutralize coercion and ascertain the correct election outcome.

The messy process of a court case may seem far less appealing than the clean technical solution provided by coercion-resistance. However, we argue that coercion-resistance is only a partial solution to the problem of coercion: the goal of a voting system is not only to output the correct outcome, but also to convince voters that this outcome is indeed proper and correct. An audit may be messy, but voters are already familiar with its mechanics and understand how the adversarial legal system serves to arrive at a fair outcome; the lay voter is far less likely to understand why cryptography is able to guarantee the fairness of an election in the presence of coercion.

Furthermore, laws are the ultimate arbiter of election propriety, so far from a disadvantage, it is inevitable and beneficial that the courts be involved in adjudicating the election outcome. It is then the purpose of the voting system to provide extensive and clear evidence to guide the court. Cryptographic voting systems are designed to satisfy the mathematician that coercion has been mitigated in a given election, but this may not be the most useful evidence for the court's purposes. Our system offers a high degree of transparency: it outputs an audit trail that includes full ballot scans, physical ballots that may be subjected to forensic analysis, the number of cancelled (and possibly coerced) ballots, and possibly the voter identities corresponding to suspect ballots (if the protocol allows for their deanonymization). All of this data can be handled in a way analogous to that of a traditional audit. This represents a significant advantage over most cryptographic voting systems, which offer very little in the way of transparency or auditability. As Benaloh (2008) mentions, audits are a complimentary approach to end-to-end verifiability and may better handle widespread attacks.

Abandoning the cryptography of coercion-resistance also allows for superior usability. In a coercion-resistant system, each voter must go through the rigamarole of a distributed registration protocol to construct a series of cryptographic credentials, must encrypt their ballots, and must submit appropriate proofs, and the voter must do this even if they are not being coerced. Our system

---

<sup>14</sup>In fact, the original paper by Juels et al. (2005) mentions that mail can be used as an untappable registration channel. This makes sense when designing an Internet voting system, when the goal may not be a system that is perfectly secure in an absolute sense, but rather a system that is no less secure than current election practice. We aim to design a system that is secure in an absolute sense, so we cannot assume mail to be untappable.

features a radically simpler ballot casting protocol completely free of cryptography. Furthermore, only coerced voters need to understand the details of the multiple-cast policy; most voters can simply cast a ballot using the first credential they receive and do not need to worry about the parts of the system that provide coercion-evidence and verifiability. In the limit where there are no coerced or malicious voters, and hence no cancelled votes, the tally is verifiable *without any cryptography*. In other words, the complexity of the coercion-evidence and verification mechanisms of the system only exhibits itself when it is necessary. In the absence of coercion, voters and administrators interact with the system in a way little different from current vote-by-mail practice.

#### 4. DISCUSSION

The motivating feature of the voting system we have described is that it publishes ballot scans, bringing transparency to the voting process by allowing for public audits of the ballot box. This is far from a novel goal, however: volunteers in Humboldt County, California, scanned ballots from two 2008 elections, citizens in Colorado have sued for the right to access ballot scans, and similar efforts are underway in other states (Adler & Hall, 2013).

While posting ballot scans in the name of transparency may seem a beneficial development in election administration, Adler and Hall (2013) compellingly argue that doing so would do more harm than good to the integrity of the electoral process. This approach to transparency is plainly untenable if ballots could be associated with the voter who submitted them. Such a violation of privacy would be illegal; the constitutions of all fifty states guarantee ballot secrecy and furthermore would allow unrestricted vote-buying and coercion. Ballot publication seems possible, however, if one ensures that the ballots are not identifiable.

The problem is that it is impossible to guarantee that a ballot is truly anonymous. The most innocuous of stray marks is enough to distinguish a ballot. Moreover, ballots in the U.S. commonly include dozens of races; a coerced voter or vote-seller may uniquely sign their ballot by voting for an agreed-upon sequence of candidates. Many states have statutes that criminalize marking a ballot in an identifiable way or invalidate the vote therein; California law specifically prohibits the publication of ballots with identifiable marks (Adler & Hall, 2013). The trouble is, of course, that there is no way for an election official to reliably determine whether a stray mark or sequence of votes was made with the intent of making the ballot identifiable. Given the impossibility of such a task, one might reasonably conclude that ballot publication cannot be done without breaking the law and undermining anonymity.

Our proposed voting system is the first paper-based system to allow ballot publication while addressing the aforementioned concerns. A voter may choose to make any ballot they cast identifiable, but they can always cast another ballot to cancel the previous vote. As we have discussed previously, this is sufficient to neutralize vote-selling and coercion.

That said, publishing ballots may do harm to the electoral process even if voters have no rational basis on which to fear privacy loss. Gerber, Huber, Doherty, and Dowling (2012) have demonstrated that voter behavior is driven by their perception of privacy, which may be quite different than their actual level of privacy. In their survey, a quarter of respondents did not believe their ballot choices were kept secret. This surprisingly high fraction suggests that voters may be unfamiliar with the regulations and procedural safeguards in place to protect their privacy. These doubts are consequential: they lead to depressed turnout (Gerber, Huber, Doherty, Dowling, & Hill, 2013a) and in some cases may influence how a voter votes (Gerber, Huber, Doherty, & Dowling, 2012). Furthermore, Claassen, Magleby, Monson, and Patterson (2012) observe that voters' perceptions of privacy are correlated with their belief in a fair election outcome. In a later work, Gerber, Huber, Doherty, Dowling, and Hill (2013b) find that postal voters are more likely to doubt the secrecy of their ballot than in-person voters. In this survey, 43% of postal voters reported that it would be not difficult at all or not too difficult to find out who [they] voted for, and a similar number reported that they thought that election officials access [their] voting records to figure out who [they] voted for. Thus, ameliorating voters' privacy concerns should be a key goal of any vote-by-mail system.

Unfortunately, the posting of ballot scans runs the risk of inflaming these concerns. Every voter will know that anyone else could look at their ballot, and might believe that someone could identify which ballot was theirs, even if they did not have reason to believe this.

Unlike in existing vote-by-mail systems, the government cannot learn how they voted (except possibly with the authorization of an election judge). While voters may not understand or appreciate the cryptography that serves to protect their privacy, they do not need to: ballot secrecy under current election administration is assured not by mathematical proof but by procedural means. Gerber, Huber, Biggers, and Hendry (2013) find that mailings reminding voters of their rights to a secret ballot are effective in assuaging voters' privacy doubts and yield a long-term increase in turnout.

Posting ballot scans also gives voters the ability to choose to give up their anonymity. If voters were to voluntarily give up their privacy in large numbers, it would further undermine confidence in the secret ballot. Moreover, if many voters denonymized their ballot, it would create social pressure for others to follow suit. As such, laws prohibiting making ballots identifiable should be kept in place for the sake of upholding the *perception* of privacy even if they are not necessary to ensure to ensure actual privacy. Publicizing these regulations on ballots and other voting materials would go a long way toward ameliorating voter concerns.

We see that publishing ballot scans may have negative consequences for the perception of voter privacy, although reminders about secrecy regulations and procedures on election materials and through mailings may in large part effectively mitigate this. The purpose of publishing ballots, however, is to give voters confidence that their ballots were received; this is an unambiguous strength of our system. This approach is especially desirable in situations where ballot transport is highly unreliable. For example, our system could significantly improve the trustworthiness of UOCAVA voting, but would do so without voters to be coerced or their privacy violated.

However, our approach is useful more generally to combat the electorate's well-founded lack of confidence in postal voting. Alvarez, Hall, and Llewellyn (2008) have found that the fraction of absentee mail voters reporting that they are very confident that their vote counted was 16% lower than the corresponding fraction for optical scan voters. These doubts are not unfounded. (Alvarez, Hall, & Sinclair, 2008) find that absentee ballots cast by mail are much more likely to be challenged or not counted than ballots cast in person. In our system, since ballots and canvassing board decisions are posted publicly, voters can be directly verify that their ballot was received intact, before the deadline, and interpreted correctly. If their ballot was challenged or invalidated, they can see this too, and if they wish they may register a dispute with the election authorities.

Postal voters have been shown to have increased concern with privacy and decreased confidence in the integrity; both of these factors have been shown to depress turnout (Alvarez, Hall, & Llewellyn, 2008; Gerber et al., 2013a). By specifically reassuring the voter in both of these areas, our system may well fulfill one of the elusive promises of voting-by-mail: unambiguously increased turnout.

We have thus described a system which attempts to address exactly those concerns about which voters care most. The central verifiability mechanism of our design—the posting of plaintext ballots—is enabled by the multiple-voting with cancellation procedure of Grewal et al. (2013). In our system, however, we postpone the encryption of the votes until after casting. By doing so, our system allows voting-by-mail, and allows publication of ballot plaintext, two novel features in a voting system with coercion mitigation.

While the proposal allows highly transparent postal voting, it falls short of a fully-verifiable postal voting scheme. It remains an interesting, and highly relevant, goal for future work to construct a system that makes further progress in balancing verifiability with usability in the postal voting setting.

#### A. INFORMATION LEAKAGE FROM TALLY AND BALLOTS

We model a ballot with  $k$  binary options as a bit-vector  $b \in \{0, 1\}^k$  (a 0 represents no mark, a 1 represents a mark), the ballot data  $M$  for  $n$  ballots is a  $k \times n$  matrix of bits, and the tally is given by a vector  $t \in \mathbb{N}^k$ . Recall that the tally is not the total number of marks for that ballot option, but the

total number of marks for that ballot option *not including cancelled ballots*. Note that the number of cancelled ballots,  $c$ , is evident from  $\mathcal{BB}$ . A solution vector is a vector  $\chi \in \{0, 1\}^n$  with entries  $\chi_i$ ,  $1 \leq i \leq n$  such that  $M\chi = t$  and  $|\chi|_1 = n - c$ . That is, a solution vector labels each ballot as either non-cancelled (contributes to the tally) or cancelled (does not contribute to the tally) in such a way as the tally of such non-cancelled ballots  $M\chi$  equals the actual tally  $t$ , and furthermore since the number of cancelled ballots  $c$  is public knowledge, the solution vector must only label  $c$  ballots as cancelled. Let  $\mathcal{S}$  be the set of solution vectors. The privacy guarantees we seek are negated when an adversary may learn with near-certainty that a particular ballot was cancelled. Thus, privacy loss occurs when an adversary may find a  $P[\chi_i = 0]$  close to 1 for a ballot of interest  $i$  (where presumably *close* means  $P[\chi_i = 0]$  considerably in excess of  $\frac{c}{n}$ , the probability one obtains knowing only the number of cancelled ballots and not the tally or ballot data). We may set  $P[\chi_i = 0] = \frac{|\{\chi \in \mathcal{S} | \chi_i = 0\}|}{|\mathcal{S}|}$ . To calculate this, one needs to find the solution set  $\mathcal{S}$  given ballot data  $M$  and a tally  $t$ .

For typical election settings many ballots, many voters we would not expect an adversary to be able to carry out this attack and violate privacy in this matter; it is left for future work to prove a privacy bound that makes such an argument rigorous. Alternatively, we can modify the voting protocol to prevent any possibility of such an attack. The notion of privacy we need is essentially that we want the output of our protocol the tally to be insensitive to which ballots we cancel. This is exactly the goal of *differential privacy*, a well-studied framework for privacy-preserving computation (Dwork, 2006). The usual method to implement a differentially-private algorithm is to add a small amount of noise to the output. While adding noise to an election seems untenable at first, notice that the amount of noise we would need to add (on the order of one vote) is negligible compared to other sources of noise in real-world elections. Furthermore, the probability that this noise would change the outcome of the election is exponentially small, and we can neglect it for all practical purposes.

We now sketch how one might add noise using a cut-and-choose protocol. Before the registration phase, the trustees generate one extra credential  $\text{cred}_{\text{noise}}$  which will be used to inject noise, and a noise source (does not have to be trusted) generates  $N$  (with  $N \sim 1000$ ) instances of random ballot data  $B_i^l$ ,  $1 \leq i \leq N$  for each race  $l$ , and posts the encryption  $\{B_i^l\}_{PK}$  of each of them to  $\mathcal{BB}$ . After the balloting phase, a trusted source of randomness (e.g., stock market data, cf. Clark, Essex, and Adams (2007), Clark and Hengartner (2010)) is used to select an  $1 \leq k \leq N$ . During tallying,  $(\text{cred}_{\text{noise}}, B_k^l)$  is then included in the mixnet and processed like any other ballot. After the tallying, the trustees jointly decrypt the other  $N - 1$  instances of random ballot data; for large  $N$ , it can be verified that  $B_k^l$  was selected randomly with high probability.

## References

- Adida, B. & Rivest, R. L. (2006). Scratch & Vote: Self-contained Paper-based Cryptographic Voting. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society* (pp. 29–40). WPES '06. ACM.
- Adler, E. S. & Hall, T. E. (2013). Ballots, Transparency, and Democracy. *Election Law Journal*, 12(2), 146–161.
- Alvarez, R. M., Ansolabehere, S., Berinsky, A. J., Lenz, G., Stewart, C., III, & Hall, T. E. (2009). *2008 Survey of the Performance of American Elections*. Caltech/MIT Voting Technology Project.
- Alvarez, R. M., Hall, T. E., & Llewellyn, M. H. (2008, July). Are Americans Confident Their Ballots Are Counted? *The Journal of Politics*, 70(03).
- Alvarez, R. M., Hall, T. E., & Sinclair, B. (2008). Whose absentee votes are returned and counted: The variety and use of absentee ballots in California. *Electoral Studies*, 27(4), 673–683.
- Alvarez, R. M., Stewart III, C., & Beckett, D. (2013). Voting Technology, Vote-by-Mail, and Residual Votes in California, 1990-2010. *Political Research Quarterly*, 66(3), 658–670.
- Aslam, J. A., Popa, R. A., & Rivest, R. L. (2007). On estimating the size and confidence of a statistical audit. In *Proceedings of the 2007 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT07)*.



- Benaloh, J. (2008). Administrative and Public Verifiability: Can We Have Both? In *Proceedings of the conference on electronic voting technology*. EVT '08. Berkeley, CA: USENIX Association.
- Benaloh, J. (2013, August). Rethinking Voter Coercion: The Realities Imposed by Technology. *Journal of Election Technology and Systems (JETS)*, 1(1), 82–87.
- Benaloh, J., Ryan, P. Y. A., & Teague, V. (2013). Verifiable postal voting. In B. Christianson, J. Malcolm, F. Stajano, J. Anderson, & J. Bonneau (Eds.), *Security Protocols XXI* (Vol. 8263, pp. 54–65). Lecture Notes in Computer Science.
- Bergman, E. (2012, February). Administering Democracy: Public Opinion on Election Reform in California. *The California Journal of Politics & Policy*, 1–24.
- Bland, S. (2014, February). Tracking Voters in Real Time in Colorado. Retrieved February 22, 2014, from <http://www.nationaljournal.com/magazine/tracking-voters-in-real-time-in-colorado-20140224>
- Brandt, F. (2006). Efficient Cryptographic Protocol Design Based on Distributed El Gamal Encryption. In D. Won & S. Kim (Eds.), *Information Security and Cryptology (ICISC 2005)* (Vol. 3935, pp. 32–47). Lecture Notes in Computer Science. Springer Berlin Heidelberg.
- Bursuc, S., Grewal, G. S., & Ryan, M. D. (2012). Trivitas: Voters Directly Verifying Votes. In *Proceedings of the Third international conference on E-Voting and Identity* (pp. 190–207). VoteID '11. Springer-Verlag.
- Calandrino, J. A., Clarkson, W., & Felten, E. W. (2011). Bubble Trouble: Off-line De-anonymization of Bubble Forms. In *Proceedings of the 20th USENIX Conference on Security*. SEC '11. USENIX.
- Chaum, D. (2004). Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy*, 2(1), 38–47.
- Chaum, D. L. (1981, February). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), 84–90.
- Chaum, D., Ryan, P. Y., & Schneider, S. (2005). A practical voter-verifiable election scheme. In *Computer Security (ESORICS 2005)* (pp. 118–139). Lecture Notes in Computer Science. Springer.
- Claassen, R. L., Magleby, D. B., Monson, J. Q., & Patterson, K. D. (2012, May). Voter Confidence and the Election-Day Voting Experience. *Political Behavior*, 35(2), 215–235.
- Clark, J., Essex, A., & Adams, C. (2007). Secure and observable auditing of electronic voting systems using stock indices. In *Canadian Conference on Electrical and Computer Engineering, 2007 (CCECE 2007)* (pp. 788–791). IEEE.
- Clark, J. & Hengartner, U. (2010). On the Use of Financial Data as a Random Beacon. *IACR Cryptology ePrint Archive, 2010*, 361.
- Clarkson, M., Chong, S., & Myers, A. (2008). Civitas: toward a secure voting system. In *Security and privacy, 2008. sp 2008. ieee symposium on* (pp. 354–368). doi:10.1109/SP.2008.32
- Dategrity Corp. (2005, May). VoteHere Announces Mail-in Ballot Tracker Audit Solution. Retrieved from <http://www.marketwired.com/press-release/votehere-announces-mail-in-ballot-tracker-audit-solution-663738.htm>
- Dwork, C. (2006). Differential privacy. In *Automata, languages and programming* (pp. 1–12). Springer.
- Elgamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4), 469–472. doi:10.1109/TIT.1985.1057074
- Essex, A. & Hengartner, U. (2012). Oblivious Printing of Secret Messages in a Multi-party Setting. In A. D. Keromytis (Ed.), *Financial Cryptography and Data Security* (Vol. 7397, pp. 359–373). Lecture Notes in Computer Science. Springer Berlin Heidelberg.
- Estonian National Electoral Committee. (2013a). Internet Voting in Estonia. Retrieved December 3, 2013, from <http://www.vvk.ee/voting-methods-in-estonia/engindex/>

- Estonian National Electoral Committee. (2013b). Statistics about Internet Voting in Estonia. Retrieved December 3, 2013, from <http://www.vvk.ee/voting-methods-in-estonia/engindex/statistics>
- Gerber, A. S., Huber, G. A., Biggers, D. R., & Hendry, D. J. (2013). Ballot Secrecy Concerns and Voter Mobilization: New Experimental Evidence about Message Source, Context, and the Duration of Mobilization Effects. Manuscript, Yale University, Department of Political Science.
- Gerber, A. S., Huber, G. A., Doherty, D., & Dowling, C. M. (2012, July). Is There a Secret Ballot? Ballot Secrecy Perceptions and Their Implications for Voting Behaviour. *British Journal of Political Science*, 43(01), 77–102.
- Gerber, A. S., Huber, G. A., Doherty, D., Dowling, C. M., & Hill, S. J. (2013a). Do Perceptions of Ballot Secrecy Influence Turnout? Results from a Field Experiment. *American Journal of Political Science*, 57(3), 537–551.
- Gerber, A. S., Huber, G. A., Doherty, D., Dowling, C. M., & Hill, S. J. (2013b). The Voting Experience and Beliefs about Ballot Secrecy. Manuscript, Yale University, Department of Political Science.
- Greenson, T. (2009, March). SOS report: Numerous deficiencies in elections software. Retrieved August 5, 2013, from [http://www.times-standard.com/ci\\_11841759](http://www.times-standard.com/ci_11841759)
- Grewal, G. S., Ryan, M. D., Bursuc, S., & Ryan, P. Y. A. (2013). Caveat Coercitor: Coercion-Evidence in Electronic Voting. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy* (pp. 367–381). SP '13. IEEE.
- Gronke, P., Galanes-Rosenbaum, E., Miller, P. A., & Toffey, D. (2008, June). Convenience Voting. *Annual Review of Political Science*, 11(1), 437–455.
- Hirt, M. & Sako, K. (2000). Efficient receipt-free voting based on homomorphic encryption. In *Advances in Cryptology (EUROCRYPT 2000)* (pp. 539–556). Springer.
- Jakobsson, M. & Juels, A. (2000). Mix and match: secure function evaluation via ciphertexts. In T. Okamoto (Ed.), *Advances in Cryptology (ASIACRYPT 2000)* (Vol. 1976, pp. 162–177). Lecture Notes in Computer Science. Springer Berlin Heidelberg.
- Jakobsson, M., Juels, A., & Rivest, R. L. (2002). Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking. In *Proceedings of the 11th USENIX Security Symposium* (pp. 339–353). USENIX Association.
- Jakobsson, M., Sako, K., & Impagliazzo, R. (1996). Designated verifier proofs and their applications. In *Advances in Cryptology (EUROCRYPT 96)* (pp. 143–154). Springer.
- Juels, A., Catalano, D., & Jakobsson, M. (2005). Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society* (pp. 61–70). WPES '05. ACM.
- Kim, E., Carlini, N., Chang, A., Yiu, G., Wang, K., & Wagner, D. (2013, August). Improved support for machine-assisted ballot-level audits. *Journal of Election Technology and Systems (JETS)*, 1(1), 88–105.
- National Conference of State Legislatures. (2013). Absentee and Early Voting. <http://www.ncsl.org/legislatures-elections/elections/absentee-and-early-voting.aspx>.
- Nixon, R. (2013, July). U.S. Postal Service Logging All Mail for Law Enforcement. Retrieved July 3, 2013, from <http://www.nytimes.com/2013/07/04/us/monitoring-of-snail-mail.html>
- Popoveniuc, S. & Lundin, D. (2007). A simple technique for safely using Punchscan and Prt Voter in mail-in elections. In *Proceedings of the 1st International Conference on E-voting and Identity (VOTE-ID'07)* (pp. 150–155).
- Rivest, R. L. (2006). The ThreeBallot voting system. Retrieved from <http://theory.csail.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf>
- Rivest, R. L. & Shen, E. (2012). A Bayesian method for auditing elections. In *Proceedings of the 2012 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE'12)*.

- Ryan, P. Y. A. (2005). A Variant of the Chaum Voter-verifiable Scheme. In *Proceedings of the 2005 Workshop on Issues in the Theory of Security (WITS '05)* (pp. 81–88). ACM.
- Saeednia, S., Kremer, S., & Markowitch, O. (2004). An efficient strong designated verifier signature scheme. In *Information security and cryptology (icisc 2003)* (pp. 40–54). Springer.
- Sharma, A., Subramanian, L., & Brewer, E. A. (2011). Paperspeckle: microscopic fingerprinting of paper. In *Proceedings of the 18th ACM conference on Computer and communications security* (pp. 99–110). CCS '11. New York, NY, USA: ACM.
- Sinclair, D. E. B. & Alvarez, R. M. (2004). Who Overvotes, Who Undervotes, Using Punchcards? Evidence from Los Angeles County. *Political Research Quarterly*, 57(1), pages.
- Smart, M. & Ritter, E. (2009). Remote Electronic Voting with Revocable Anonymity. In *Proceedings of the 5th international conference on information systems security* (pp. 39–54). ICISS '09. Berlin, Heidelberg: Springer-Verlag.
- Southwell, P. L. (2004). Five Years Later: A Re-Assessment of Oregon's Vote by Mail Electoral Process. *PS: Political Science and Politics*, 37(1), 89–94.
- Stark, P. B. (2008). Conservative statistical post-election audits. *Annals of Applied Statistics*, 2(2), 550–581.
- Stark, P. B. (2009). CAST: Canvass audits by sampling and testing. *IEEE Transactions on Information Forensics and Security*, 4(4), 708–717.
- Stewart, C., III. (2010). Losing votes by mail. *NYU Journal of Legislation & Public Policy*, 13, 573.
- Stiefbold, R. P. (1965, June). The Significance of Void Ballots in West German Elections. *American Political Science Review*, 59, 391–407.
- Zagrski, F., Carback, R., Chaum, D., Clark, J., Essex, A., & Vora, P. L. (2013). Remotegrity: Design and Use of an End-to-End Verifiable Remote Voting System. *IACR Cryptology ePrint Archive*, 2013, 214.

#### ACKNOWLEDGMENTS

Foremost, the authors would like to thank Kenneth Hung, as the ideas presented here largely arose in discussions with him. This work benefitted greatly from careful and detailed comments on earlier drafts by Katrina Ligett, Barath Raghavan, Ron Rivest, Jonathan Katz, Charles Stewart III, and the anonymous referees.

Received August 2013; revised June 2014; accepted July 2014