Scantegrity: End-to-End Voter-Verifiable Optical-Scan Voting

Scantegrity is a security enhancement for any optical scan voting system. It lets voters verify that their ballots were correctly recorded and counted, but doesn't change how voters mark their ballots.

David Chaum

Aleks Essex
University of Ottawa

Richard Carback and Alan Sherman
University of Maryland, Baltimore County

Jeremy Clark
University of Waterloo

Stefan Popoveniuc and Poorvi Vora
The George Washington University

Voter confidence in the US electoral process is eroding as a steady stream of reports continue to expose fundamental security flaws in certified electronic voting machines.[1,2] Similar voting technology is used outside of the US, and resistance to electronic voting has spread to other democracies. Although proposals such as stricter design standards, more open systems (preferably open source), and independent verification methods (such as paper audit trails) are improvements, they don't go far enough. In any voting system, with or without an electronic component, the core security problem is *chain of custody*. An attacker who breaks chain of custody could stuff the ballot box, delete or switch votes, or add votes to contests that the voter left empty. Whether the attacker accomplishes this by inserting malicious code or altering paper ballots, such attacks go undetected even with a manual vote recount.

Recently proposed end-to-end (E2E) verification voting systems have focused on minimizing voting systems' reliance on chain of custody.[3] These E2E systems typically provide cryptographic checks indicating that ballots have been recorded as cast and tallied as recorded. Voters can check that their votes are recorded accurately using a receipt, and any observer can verify that the tally is correctly constructed, all without compromising ballot secrecy. In these systems, any chain-of-custody break causes a detectable alteration of the public record. In particular, erroneous voting machine software or voting machine malfunction doesn't dilute the voting systems' integrity. However, using these systems in real elections has been a challenge. They typically require a special type of ballot format—for example, Punch-scan ballots[4] require two sheets of paper, and Prêt à Voter ballots[5] randomize candidate name order.

The Scantegrity voting system combines E2E systems' cryptographic ideas with the familiarity of a widely used vote-counting system. It thus provides the strong security guarantees of E2E systems but is unobtrusive to the voter, has a minimal cost for wide-scale deployment, and doesn't interfere with existing procedural requirements such as paper audit trails and manual recounts. Scantegrity is designed for use with optical scan voting systems, which are the most widely used election technology in the US and are being adopted in other countries as well.[6] Scantegrity can be readily deployed in precincts with existing optical scan systems because it adds minimal requirements to the underlying optical scan process and doesn't introduce any new polling place equipment. It only requires extra information to be printed on the ballots during production and system access to the raw scan results after the election. In summary, Scantegrity minimally impacts election procedures and is the first independent E2E verification mechanism that preserves optical scan as the underlying voting system and doesn't interfere with a manual recount.

Independent E2E verification voting systems

E2E systems, sometimes called receipt-based or universally verifiable voting systems, don't derive security from any specific type of voting equipment. Instead, they generally produce an encrypted representation of

ballot choices that functions as a receipt. This receipt doesn't reveal the voter's identity or choices, so the voter can take home a signed or stamped copy of his or her receipt. Election officials publicly post the receipts of all the ballots they received, and voters can check this record to see that their ballot was included and is unmodified. If a voter's receipt doesn't appear or was modified, the signature or stamp on the receipt gives the voter proof of a discrepancy that he or she can use to dispute the record.

Election officials can only decrypt the receipts to recover the final tally by using a method that hides which decrypted ballots correspond to which receipts. They then perform a mandatory audit to prove mathematically that the public record was decrypted properly and that the votes were unmodified—whether by a software error, the election officials, a hacker, or some other entity. Any independent party can author the software used to verify the election outcome using a public specification. Under this model, we expect that several independent entities will make their software tool version freely available to the public, mitigating the issue of software error.

Other E2E systems include VoteHere's MarkPledge,[7,8] Votegrity,[9] Punchscan,[4] Prêt à Voter,[5] and Voter Initiated Auditing.[10] Scantegrity is closely related to Punchscan, but unlike its predecessors, it can also serve as an add-on to existing voting technology without interfering with the underlying system's tabulation procedures or paper audit trails. It also uses a new, simpler mechanism to replace the basic vote tallying mechanism of other E2E schemes.

Voter experience

The voter experience in Scantegrity is identical to that of regular optical scan systems (that is, the voter marks the optical scan ballot and feeds it into an optical precinct scanner), except the voter can take home a privacy-preserving receipt. To create the receipt, a voter tears off a perforated corner of the ballot, called a *ballot chit*, that contains a serial number. In addition, as Figure 1 shows, the voter writes down the randomly assigned code letter listed next to the selected candidate. Note that in the ballot in Figure 1, the letter A is beside Bob's name, but it might be beside Alice's name on other ballots. Thus, knowing that someone voted for a particular code letter doesn't tell you which candidate that person voted for.

After the underlying optical scan system tallies the election results, election officials post a public record containing the Scantegrity serial numbers and chosen code letters of all the scanned ballots, but not the candidate associated with the letter on each ballot. Voters can retrieve this public record, look up their serial number, and verify that the code letters they wrote match those in the posted record. Voters can also make a copy of the receipt and give it to third parties, who can also check the public record. The more receipts checked, the higher the chance of detecting a problem with the public record.[11]

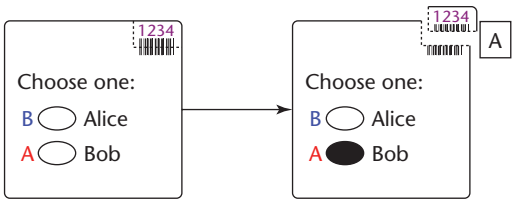

Figure 1. The Scantegrity ballot. Scantegrity uses an optical scan ballot with randomly assigned code letters next to each choice. The perforated chit in the corner contains a serial number written in human- and computer-readable forms.

Correct letters indicate to voters that the officials properly scanned and recorded their votes. However, if the public record contains a letter that's different from what a voter recorded, the voter can challenge the record through a dispute-resolution process.

Dial "0" for independent verification

Anyone can use the public record to verify the tally—that is, that the results were counted as recorded. Tally verification is challenging because the system must not directly reveal the links between code letter and candidate to preserve ballot secrecy.

Although Scantegrity isn't the first system to provide counted-as-recorded integrity verification,[4,5,7–9] its solution is the simplest. The importance of solution simplicity can't be overemphasized in voting—it lets the widest possible audience understand how the voting system works.

Some E2E solutions use a mix network[12] to create an anonymous but verifiable link between receipt and vote. The mix network applies a cryptographic operation at each node to obscure the path of messages through the network. This is especially important for identifiable (that is, unique) data such as email messages. Some E2E systems provide encrypted information on the ballot receipt (this information is sometimes called an "onion") to let the mix network perform the correct cryptographic operations to count the vote correctly. Punchscan, Scantegrity's direct ancestor, uses a simplified two-stage mix network with efficient cryptographic operations. It also doesn't use an onion on the ballot.

Election data doesn't necessarily need to be explicitly encrypted. Under the familiar plurality voting system (also known as "first-past-the-post"), voters express their intent by making a mark beside their chosen candidate. A cast ballot therefore can be ex-

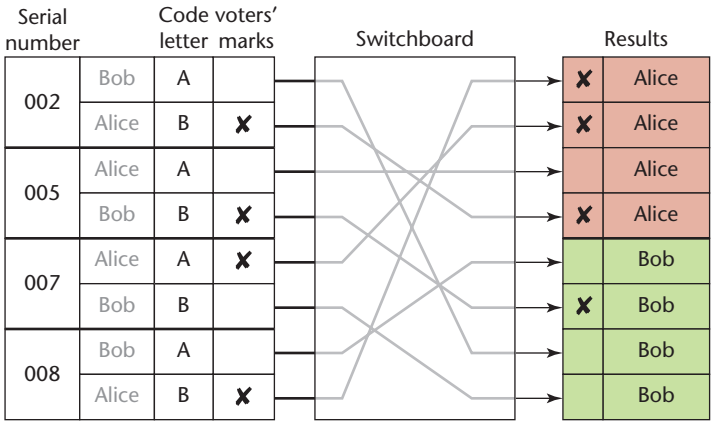

Figure 2. The switchboard. Marks beside code letters are routed to marks beside candidates using a random and obliviously generated circuit-switched network.

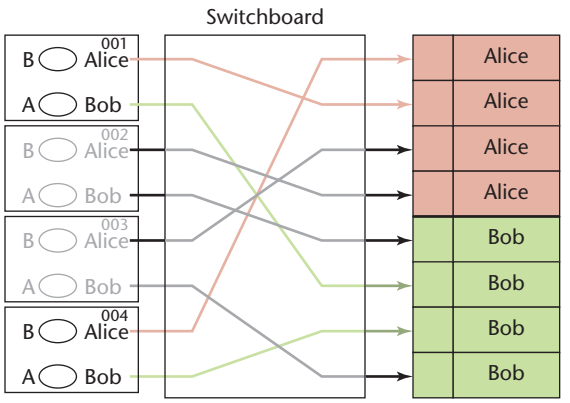

Figure 3. Ballot printing audit performed before an election. The figure shows revealed ballots 001 and 004, the association of code letters, and their connections through the switchboard. This information is made publicly available, and any independent party can see a mark for Alice or Bob would have been correctly registered as a vote for Alice or Bob, respectively. Once revealed, these ballots aren't used in the election.

pressed in terms of a specific collection of marked or unmarked regions. If treated individually, the states of these markable regions aren't unique, so they don't need to be encrypted as they pass through an anonymizing network. Instead of using a mix network architecture, we achieve the same anonymity properties through a simpler process—a secret permutation of the states of the markable regions (akin to shuffling a deck of cards). Thus, Scantegrity uses the permutation to recover the vote while hiding the link between serial number and vote.

The *switchboard* (see Figure 2) is a collection of circuits established between specific markable regions on ballots (marked or not marked) and a particular candidate (voted for or not voted for). The *trustee workstation* transmits the state of each markable region on each ballot in the election through the switchboard to votes for the corresponding candidates in the election results.

Finally, the public must be able to verify independently that marks are being transmitted to the correct candidate without exposing both of the circuit's end points (receipt and vote).

Auditing the switchboard

To ensure voter confidence in the switchboard's ability to produce a correct tally, Scantegrity must reveal some information for verification purposes. Initially, when election trustees create the ballots and switchboard, they commit to this secret information by using a cryptographically secure bit-commitment scheme.[4,13]

Before the election, election officials generate and publish these commitments, letting independent entities verify that no one could have simply "cooked up" the secret data revealed later on during the audit process. The verification requires publicly revealing some secret data and verifying its correctness against the committed data. We reveal secret information in two ways: reveal the full secret and then discard it from use in the election, or reveal partial information that's sufficient for checking, but that doesn't reveal anything about the secret.

Figure 3 illustrates the first technique, which auditors use to verify the correctness of the association between code letter and candidate in the switchboard. Before the election, auditors randomly choose half of the ballots to be revealed publicly, along with their serial numbers and connections through the switchboard. Those performing this printing audit can ensure that the path through the switchboard for each candidate on each of the revealed ballots leads to a vote for the correct candidate in the results. They then destroy these ballots. If they chose the ballots fairly and randomly, the public has a high level of assurance that the remaining sealed ballots are printed and routed correctly. Voters can also audit the printing themselves by keeping a ballot they receive once it's marked as "spoilt."

After the polls close, we use the second technique (illustrated in Figure 4) to audit the switchboard. If we segment the switchboard into two randomly generated circuit-switched networks, revealing a link in one of the networks doesn't reveal the full connection. Voters' marks travel through the first network and are recorded in an intermediary location. The marks in the intermediary position continue through the second network to their final place in the results table. For each intermediary position, auditors challenge the election trustees to reveal either the link to it through the first network or the

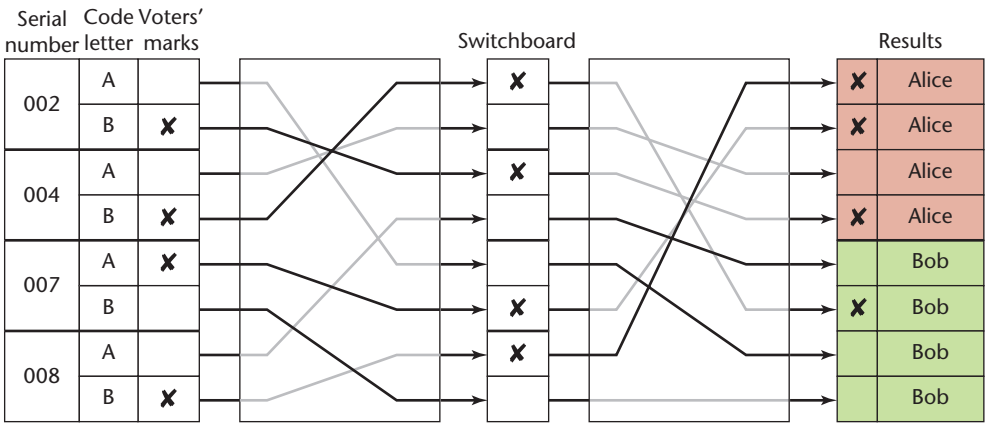

Figure 4. Mark audit performed after the election. Ballot 002 shows a mark for the candidate with code letter B and that this mark was correctly recorded in the intermediate position. Likewise, the first vote for Alice in the results table was correctly copied from the intermediary position. Knowing only one link doesn't reveal the connection between code letter and candidate, preserving the receipt's privacy.

link from it through the second network, but never both. Thus, the connection between a recorded receipt and its position in the final results table is never revealed. For each of these links, anyone can publicly verify that a mark (or absence of mark) traveled through the link unchanged. In this way, observers can be assured that the remaining secret links also routed marks correctly. To increase the audit's statistical certainty, trustees may be mandated to use multiple instances of the switchboard with different random links.

These print and mark audits, in conjunction with the receipt check, provide the verification process's end-to-end nature: integrity is ensured from ballot printing through to the final tally.

Although these audits are conceptually simple to perform, any nontrivial-sized election would warrant the use of a software audit tool to perform these repetitive checks quickly. The software tool is intended to be open source, exceptionally easy to use, and universally available to anyone for free. Concerned parties can code their own independent version following a published specification.

System architecture

Figure 5 shows how Scantegrity interfaces with the optical scan election process. The election authority—a collection of election trustees—uses a workstation on three separate occasions to compute all the information Scantegrity needs. This set of meetings represents Scantegrity's three core processes:

- Before the ballots are printed, election trustees use the workstation to compute the serial number and code letters to add to the optical scan ballots as well as generate the switchboard connections. They cryptographically commit to this (secret) data and post the commitments publicly.
- After the marked ballots are scanned on election day, election trustees give the *electronic ballot images* (EBIs) to the Scantegrity system. They post the code letters and corresponding voter-created marks made on each ballot to the public record, which voters can compare to their receipts.
- After the election results are tabulated and published, auditors challenge the election trustees to open one half of the switchboard for each marking region to prove that they counted the ballots faithfully.

Using a workstation, the officials can regenerate all the data needed for each meeting from their passphrases, preventing the need to physically store any sensitive election data. Trustees secure the workstation by removing any persistent data storage and boot the open source operating system and software from a self-contained medium that can undergo attestation by anyone present both before and after its use.[4] Fortifying the workstation protects voter privacy; the election's integrity is unconditional, and thus independent of the workstation's trustworthiness.

Resolving disputes

Because there's no control over what voters write on their receipt chits, they could write the wrong code letter. Then, when checking the official record, the voter will find a discrepancy. Officials need a voter-verifiable method to determine whether the discrepancy is the result of an incorrectly written letter or a scanner error, or malfeasance.

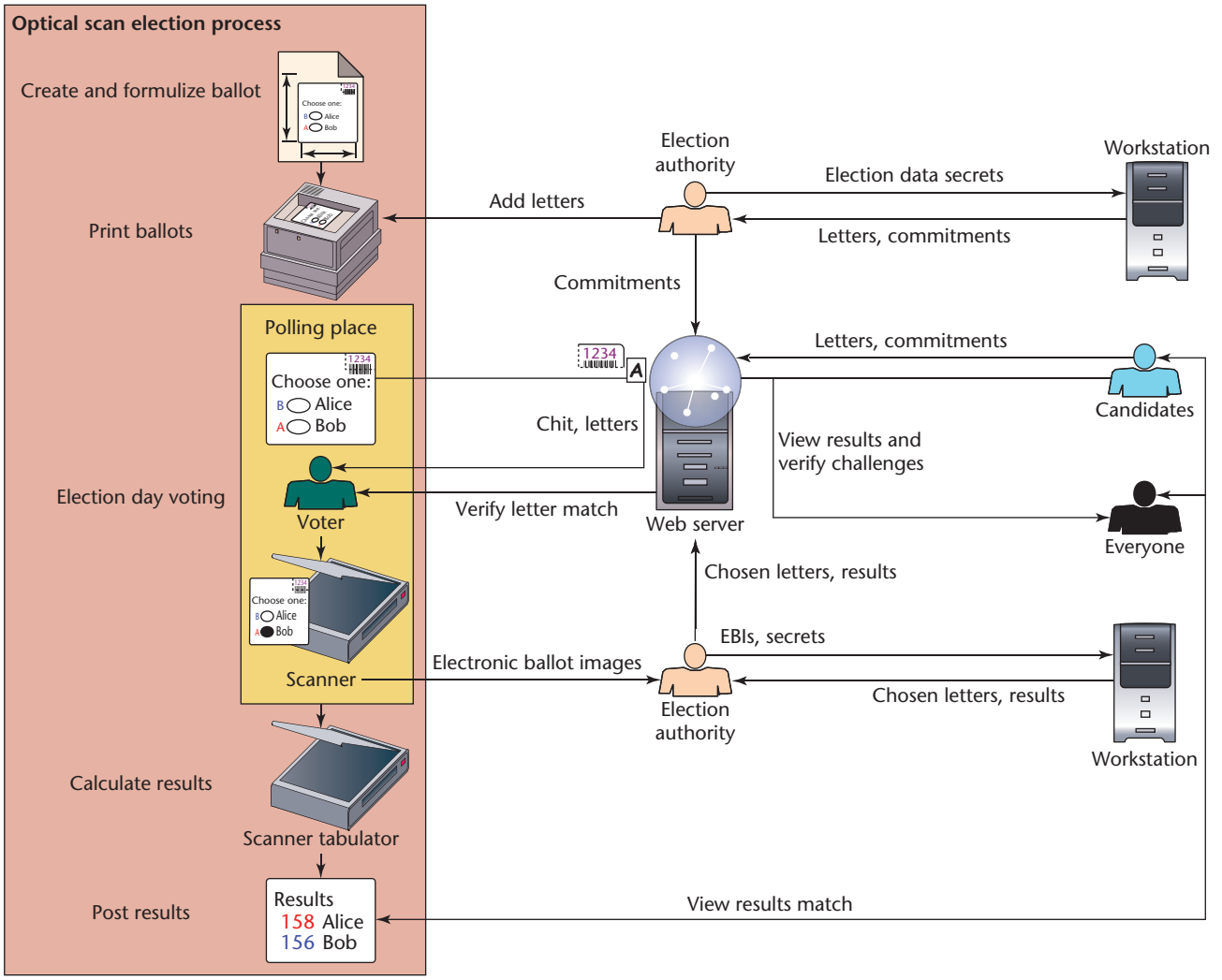

Figure 5. Overall election process with Scantegrity. Scantegrity inserts itself into the traditional optical scan election process, creating a separate mechanism for independent universal verification of election results. Election trustees use a workstation to create letters and add them to the ballots. After voting, they use the workstation again, reading the electronic ballot images (EBIs) to interpret marks on each ballot and post the chosen letters. They use the workstation a third time to respond to audit challenges, and everyone can check the responses to be sure the results tallied properly.

Figure 6 illustrates the receipt-dispute-resolution protocol. This is a two-step process that preserves ballot secrecy. First, an election official retrieves the original ballot and puts it in a privacy sleeve that reveals only the ballot's serial number, not its contents. If necessary, forensic analysis could be performed to match the chit fibers to the ballot.

Second, officials must show the code letter marked on the ballot without revealing the corresponding candidate. The official observably moves the ballot to a second privacy sleeve that will show the marks for the disputed race but not the serial number. The official notes the marked letter's position and drops the sleeve into an empty lottery-style hopper. The official then collects a set of dummy ballots with the same code letter marked for each of the other candidates, puts them in similar privacy sleeves, and drops them into the hopper.

After tumbling the hopper, the election official retrieves each privacy sleeve envelope and places it in plain view. Because the ballot was already matched to the chit, it will be in this collection. Thus, the election officials have successfully demonstrated the code letter voted for without revealing the candidate voted for. Anyone can then compare the single code letter (marked on all the shown ballots) to the public record. After all disputes are settled, everyone can assume that the public record of chosen letters is correct, and that

no ballots were lost. If necessary, officials recompute the results from the corrected public record.

We've developed a more efficient dispute-resolution procedure that doesn't require physical interaction or forensic analysis.[14]

Implementation

We created a Java-based software implementation and merged it with the open source Punchscan codebase. Our software is general enough to author ballots of both styles, even allowing an election to mix Punchscan ballots with Scantegrity ballots. The software takes a ballot layout in PDF format as input and produces a multiple-page PDF document of the ballot collection with letters and serial numbers inserted.

We tested our implementation on an Intel P4 1.73 GHz laptop, simulating the 2000 Polk County Florida general election of 32 contests with an average of 3.2 candidates per contests and 200,000 ballots cast. Under this scenario the election trustees could produce the necessary Switchboard audit data in under 4 minutes, with which the voters could independently verify the election tally in under 2 minutes.

Security considerations

Scantegrity offers a level of integrity not found in conventional voting systems. As with any security system, Scantegrity's security properties depend both on its technology and its procedural protections. However, a few security threats could arise during a Scantegrity election.

First, a coercer might attempt to collect ballot chits and match them to marked ballots. A corrupt election official, for example, might have sufficient ballot access to attempt this attack. This situation isn't significantly different from an attack on the underlying optical scan system. A coercer with access to ballots can scan the ballots for fingerprints. Alternatively, an attacker might be able to coerce a voter to choose a unique write-in candidate or mark the ballot in a unique way.

Second, without such access, a coercer can still force a voter to choose a particular letter on the ballot, creating a random vote. In this case, a voter can fight back by spoiling a ballot until he or she receives a ballot with the letter next to the desired candidate. Alternatively, voters could exchange receipts at random, as in the Farnel voting system,[15] or each voter could give the receipt to a trusted third party to check. However, forcing a random vote is similar to forcing the voter not to vote at all.[16]

Third, an attacker might attempt to inject optical scan ballots where the candidates or letters have been printed out of their intended order. Scantegrity avoids this attack by letting voters optionally spoil the ballot they receive before they see the information on it and take it home for later checking. By spoiling a ballot, all of its corresponding commitments will be revealed, and the voter can check that the ballot was printed properly. Misprinting can hurt attackers' chances of success, because they don't know which voter will get the altered ballot or for whom that voter will vote.

Because Scantegrity provides both E2E integrity and a traditional voter verifiable paper trail, it's more likely than a purely cryptographic system to meet requirements of a human-readable paper record of votes cast.

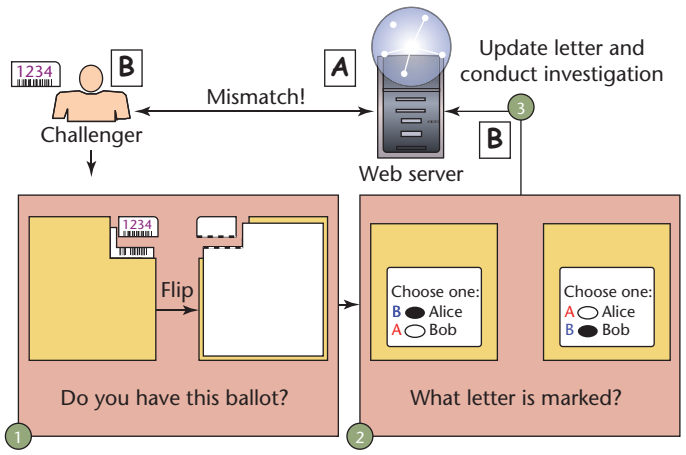

Figure 6. Dispute resolution. The election authority proves to the challenger that the ballot belonging to the chit is present and shows the letter marked on that ballot for each race. If the letter doesn't match the public record, election officials change the public record and conduct an investigation.

For democracy to stay strong, it must vigorously keep apace with the emerging vulnerabilities and possibilities of information technology—especially for its core mechanism. Scantegrity, with its simplicity, low cost, and low risk, is ready to take on the challenge and restore voter confidence. □

References

1. S. Garera and A.D. Rubin, "An Independent Audit Framework for Software Dependent Voting Systems," *Proc. 14th ACM Conf. Computer and Comm. Security* (CCS 07), ACM Press, 2007, pp. 256–265.
2. A.J. Feldman, J.A. Halderman, and E.W. Felten, "Security Analysis of the Diebold Accuvote-ts Voting Machine," *Proc. Usenix Accurate Electronic Voting Technology on Usenix/Accurate Electronic Voting Technology Workshop* (EVT 07), Usenix Assoc., 2007, p. 2.
3. US Election Assistance Commission, "2005 Voluntary Voting System Guidelines (VVSG)," Dec. 2005; www.eac.gov/voting%20systems/voting-system-certification/2005-vvsg.

4. A. Essex et al., "The Punchscan Voting System: VoComp Competition Submission," *Proc. 1st Univ. Voting Systems Competition* (VoComp), 2007; http://punchscan.org/vocomp/PunchscanVocompSubmission.pdf.
5. D. Chaum, P.Y. Ryan, and S.A. Schneider, "A Practical, Voterverifiable, Election Scheme," tech. report series CS-TR-880, School of Computer Science, Univ. of Newcastle upon Tyne, 2004.
6. Election Data Services, "2006 Voting Equipment Study," Oct. 2006; www.edssurvey.com/index.php?content=ves06n.
7. C.A. Neff, "Practical High Certainty Intent Verification for Encrypted Votes," 14 Oct. 2004; www.votehere.com/documents.php.
8. C.A. Neff, "Verifiable Mixing (Shuffling) of ElGamal Pairs," 21 Apr. 2004; www.votehere.com/vhti/documentation/egshuf-2.0.3638.pdf.
9. D. Chaum. "Secret-Ballot Receipts: True Voter-Verifiable Elections," *IEEE Security & Privacy*, vol. 2, no. 1, 2004, pp. 38–47.
10. J. Benaloh, "Ballot Casting Assurance via Voter-Initiated Poll Station Auditing," *Proc. 2007 Usenix/ACCURATE Electronic Voting Technology Workshop* (EVT 07), Usenix Assoc., 2007, p. 14.
11. J.A. Aslam, R.A. Popa, and R.L. Rivest, "On Estimating the Size and Confidence of a Statistical Audit," *Proc. Usenix/Accurate Electronic Voting Technology on Usenix/Accurate Electronic Voting Technology Workshop* (EVT 07), Usenix Assoc., 2007, p. 8.
12. D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Comm. ACM*, vol. 4, no. 2, Feb. 1981, pp. 84–90.
13. A. Kent, "Unconditionally Secure Bit Commitment," *Physical Rev. Letters*, vol. 83, no. 7, Aug. 1999, pp. 1447–1450.
14. D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. Rivest, P. Ryan, E. Shen, A. Sherman, "Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems Using Invisible Ink Confirmation Codes," submitted.
15. A.J. Devegili, "Farnel: Uma Proposta de Protocolo Criptográco para Votação Digital," Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, Brazil, 2001.
16. S. Popoveniuc and J. Stanton, "Buying Random Votes Is as Hard as Buying No-votes," Cryptology ePrint Archive, report 2008/059, 2008; http://eprint.iacr.org/2008/059.pdf.

***David Chaum**, widely recognized as the inventor of electronic cash and techniques that more generally let individuals protect their privacy in interactions with organizations, has also made fundamental contributions related to the theory of cryptography. He has taught, led a crypto research group, launched several conferences as well as international projects, and founded DigiCash and the International Association for Cryptologic Research (IACR). Chaum has a PhD in computer science from the University of California, Berkeley.*

***Aleks Essex** is a PhD student at the University of Ottawa's School of Information Technology and Engineering. He is a member of the Punchscan team, which won first place at the 2007 intercollegiate voting systems competition, VoComp. His research interests include electronic voting, information security, cryptography, and engineering design. Essex has a master's degree in electrical engineering from the University of Ottawa. Contact him at aesse083@site.uottawa.ca.*

***Richard Carback** is a PhD student at the University of Maryland, Baltimore County in the Department of Computer Science and Electrical Engineering. He is a member of the Punchscan team, which won first place at the 2007 intercollegiate voting systems competition, VoComp. Carback has an MS in computer science from UMBC. Contact him at carback1@umbc.edu.*

***Jeremy Clark** is a PhD student at the University of Waterloo's Center for Applied Cryptographic Research. He is a member of the Punchscan team, which won first place at the 2007 intercollegiate voting systems competition, VoComp. His research interests include electronic voting, online anonymity networks, usability, economics of information security, and cryptography. Clark has an MS in electrical engineering from the University of Ottawa. Contact him at j5clark@cs.uwaterloo.ca.*

***Stefan Popoveniuc** is a PhD student at George Washington University, working on computer security and privacy in general and on electronic voting in particular. He is a member of the Punchscan team, which won first place at the 2007 intercollegiate voting systems competition, VoComp. His current research interests include the relations between the various properties that voting systems have, trying to establish a framework that would allow election officials to take informed decisions when purchasing voting system to use for their special needs. Popoveniuc has a BS in computer science from Politechnical University, Bucharest, Romania. Contact him at poste@gwu.edu.*

***Alan Sherman** is an associate professor of computer science at the Department of Computer Science and Electrical Engineering at the University of Maryland, Baltimore County. He is a member of the National Center for the Study of Elections at UMBC. His research interests include high-integrity voting systems, information assurance, cryptology, discrete algorithms. Sherman has a PhD in computer science from MIT. Contact him at dralansherman@starpower.net.*

***Poorvi Vora** is an assistant professor at the Department of Computer Science at George Washington University. Her research interests include privacy, cryptology, electronic voting, and game theory. Vora has a PhD in electrical engineering from North Carolina State University. Contact her at poorvi@gwu.edu.*