Summative Usability Assessments of STAR-Vote: A Cryptographically Secure e2e Voting System That Has Been Empirically Proven to Be Easy to Use

Claudia Ziegler Acemyan, **Philip Kortum**, **Michael D. Byrne**, and **Dan S. Wallach**, Rice University, Houston, Texas

Background: From the project's inception, STAR-Vote was intended to be one of the first usable, end-to-end (e2e) voting systems with sophisticated security. To realize STAR-Vote, computer security experts, statistical auditors, human factors (HF)/human-computer interaction (HCI) researchers, and election officials collaborated throughout the project and relied upon a user-centered, iterative design and development process, which included human factors research and usability testing, to make certain the system would be both usable and secure.

Objective: While best practices in HF/HCI methods for design were used and all apparent usability problems were identified and fixed, summative system usability assessments were conducted toward the end of the user-centered design process to determine whether STAR-Vote is in fact easy to use.

Method and Results: After collecting efficiency, effectiveness, and satisfaction measurements per ISO 9241-11's system usability criteria, an analysis of the data revealed that there is evidence for STAR-Vote being the most usable, cryptographically secure voting system to date when compared with the previously tested e2e systems: Helios, Prêt à Voter, and Scantegrity.

Conclusion and Application: STAR-Vote being one of the first e2e voting systems that is *both highly usable and secure* is a significant accomplishment, because tamper-resistant voting systems can be used in U.S. elections to ensure the integrity of the electoral process, while still ensuring that voter intent is accurately reflected in the cast ballots. Moreover, this research empirically shows that a complex, secure system can still be usable—meaning that implemented security is not an excuse for poor usability.

Keywords: user-centered design (UCD), system usability, usable security, end-to-end (e2e) voting systems

Address correspondence to Claudia Ziegler Acemyan, Psychological Sciences Department, Rice University, 6100 Main Street, MS-25, Houston, TX 77005, USA; e-mail: claudiaz@rice.edu.

HUMAN FACTORS
Vol. XX, No. X, Month XXXX, pp. 1–24
DOI: 10.1177/0018720818812586

INTRODUCTION

The 2016 U.S. presidential election highlighted voters' concerns about the integrity of elections. Based on reported accusations of rigged elections, voting officials, voters, candidates, and security experts worried that nefarious, outside attackers might attempt—or already had attempted—to tamper with the voting systems to alter election outcomes. Although these concerns have not been proven, both the Federal Bureau of Investigation and the Central Intelligence Agency have indicated that a foreign power did indeed try to influence the election (Entous & Nakashema, 2016). Further, security researchers from around the world have been able to demonstrate their ability to hack into and tamper with voting systems that are currently in use (e.g., Calandrino et al., 2007; Halderman & Teague, 2015; Langone, 2018). This series of events and official statements highlights the need for voting systems to be highly secure and tamper resistant.

An essential problem is that complex security mechanisms can significantly impact the usability of the system, sometimes to the point of making the system nearly impossible to use (Tognazzini, 2005). A ubiquitous example is password authentication, in which computer users are required to develop and use their own password to access a system. Strong passwords are long, randomly generated, use a combination of upper and lower-case letters, include numbers and special characters, do not include words found in a dictionary, and are not reused across sites or over time (Cazier & Medlin, 2006). In addition, passwords should be changed frequently (Scarfone & Souppaya, 2009). Although these complex password rules are good for security (Summers & Bosworth, 2004), they are terrible for usability. Human cognition is not well suited to having to remember and recall dozens of lengthy, complex

passwords that frequently change (Stanton & Greene, 2014), much less having to do so separately for each web site with which a user might interact. Consequently, users are forced to develop strategies that make secure passwords easier to use—from writing down passwords on sticky notes that are placed on monitors so the information can be easily accessed at the time it is needed (Stanton, Stam, Mastrangelo, & Jolton, 2005) to using the same password across sites so that only one difficult password ever has to be invented and remembered. Although password users who develop adaptive strategies are rationally acting to enhance their ability to effectively and efficiently use the system, their behavior ultimately compromises security. This struggle to make secure systems also usable (including the e2e voting systems described in this article) has long been acknowledged by researchers from numerous disciplines who have studied aspects of "usable security" (e.g., Acemyan, Kortum, Byrne, & Wallach, 2014, 2015a, 2015b; Balfanz, Durfee, Grinter, & Smetters, 2004; Cranor & Garfinkel, 2005; Garfinkel & Lipford, 2014; Norman, 2010; Payne & Edwards, 2008; Sasse, Brostoff, & Weirich, 2001).

Voting System Security and Usability

Election integrity is the foundation of a democratic society. For this reason, voters' intentions must be accurately recorded on anonymous ballots that are then cast and tabulated. Security has always played a role in this process. When common paper ballots are used, ballot boxes are sealed, transported, and opened for counting while being constantly monitored by both election authorities and representatives from all political parties. After the 2000 presidential election recounts in Florida and the recognition that confusing ballots can alter elections (Wand et al., 2001), the U.S. government invested over $3.5 billion through the Help America Vote Act (2002) to modernize voting systems. The result has been a myriad of new computer voting systems, which are touted to be accurate and reliable, but have been proven to be inadequately secure and consequently extremely vulnerable to attacks (e.g., Balzarotti et al., 2010; Bishop, 2007; Epstein, 2015; Feldman, Halderman, & Felten, 2007; Proebstel et al., 2007).

As part of this recognition that confusing ballots can have adverse impacts on elections, researchers began to examine the overall usability of voting systems in an attempt to understand and quantify the impact of various interfaces. Some of this research has focused on specific elements of the ballot that impact usability, such as review screens (Campbell & Byrne, 2009a) and straight party voting interfaces (Campbell & Byrne 2009b). Overall system usability has also been a major focus. Usability assessments have been conducted in the context of traditional voting systems such as paper, punch cards, lever machines (Byrne, Greene, & Everett, 2007), and electronic voting systems (Bederson, Lee, Sherman, Herrnson, & Niemi, 2003; Conrad et al., 2009; Everett, Greene, Byrne, & Wallach, 2008; Laskowski, Autry, Cugini, Killam, & Yen, 2004). Some forward-looking research from this body of work that responded to the early 2000 election even hinted at the need for both secure systems that people could use in the areas of authentication (Paul, Evans, Rubin, & Wallach, 2003) and end-to-end security protocols (Sandler, Derr, & Wallach, 2008).

While usability is a major ongoing concern in voting, there is growing recognition that there are significant security vulnerabilities in these existing voting technologies. Due to the security flaws associated with the electronic voting machines of the 2000s combined with the fact that these machines, now more than a decade old, are nearing the end of their service lifetime, researchers from around the world have been developing the next generation of tamper-resistant, end-to-end (e2e) voting systems (e.g., Adida, 2008; Chaum, Essex, et al., 2008; Ryan, Bismark, Heather, Schneider, & Xia, 2009). E2e voting systems are voting methods that incorporate sophisticated cryptographic schemes, which make them difficult to hack. Moreover, if they are ever tampered with, they are designed so that malicious attacks can be detected through system auditing techniques, which can be carried out by voters, election officials, or any interested third party. As a result, an attacker cannot (easily) alter voters' selections and/or election outcomes without detection.

Even though e2e systems are intended to be more secure than traditional voting systems such as hand-marked paper ballots, the enhanced security comes at a cost. The voter procedures, which contribute to making these systems highly secure, also render them extremely difficult to use. Research by Acemyan and colleagues (2014) has shown that three of the most prominent e2e voting systems that have been used in real elections are so hard to vote with that about 50% of voters cannot cast votes with them, and many of these voters fail to recognize that their votes were not cast—meaning they would not know to ask for help or tell election officials. Moreover, it takes people longer to vote with them than with traditional paper ballots and electronic voting machines.

One of the three systems tested in Acemyan and her colleagues' studies was Helios, an online, computer-based voting system (Adida, 2008). To vote in an election, voters made their selections one race at a time on a web interface, then they reviewed their entire ballot on a review screen. Instead of being able to simply cast their ballot at this point, as is done on commonly used electronic voting machines, voters had to complete two different steps. First, if they wanted to be able to track their ballot, they had to hand-transcribe or print a 43-character "smart ballot tracker"—made up of numbers, upper and lowercase letters, and special characters—that was unique to their ballot. However, the voters were never told that the purpose of this smart ballot tracker is for the optional verification process that can take place after voting. Second, voters had to login to Helios using a third-party e-mail account (e.g., Gmail). Once they successfully logged in, then they were allowed to cast their ballot, which was encrypted after completing it. (This step does not take place at the beginning of the voting process because then an unencrypted ballot revealing voters' choices could potentially be associated with the user's e-mail account.) Only after completing all of these steps are their votes recorded by the system. If voters wanted to verify that the system received their ballot, they had to navigate to Helios' verification website, find their election's ballot tracking center, and then find their ballot tracker within a long list. Alternately, the voters could click on a link that was e-mailed to them and view their smart ballot tracker displayed in isolation on a web page.

When Acemyan and her colleagues studied the usability of Helios with both summative and formative methods (Acemyan et al., 2014; 2015a, 2015b), numerous usability problems were identified. To give a few examples, voters did not cast their ballot because they thought they finished voting after reviewing their ballot on the final review screen and then reading on the next screen that their ballot had been saved by the system and encrypted (which is different than being cast and recorded by the system). Others had trouble logging into their e-mail account from the Helios interface, so they were never able to get to the final vote casting step. As for the verification process, participants generally did not know what it meant to check on their vote or verify because the system did not give any explanation for this novel option that is not offered by current voting systems. Despite the numerous errors made by participants while trying to vote and verify with the system, it was found to be more usable than the other two tested e2e systems: PaV and Scantegrity. For the complete analysis resulting from the formative usability data collection methods, refer to Acemyan et al. (2015a); for photos of the Helios interface, refer to Acemyan et al. (2014); for details about the system's cryptography, refer to Adida, 2008. Please note that this article does not assess Helios' or any of the e2e systems' security as the focus is on system usability and the voters' perspectives.

The second system tested by Acemyan and her research group was Prêt á Voter (PaV; see Acemyan et al., 2014). PaV is a paper-based system in which the ballot is composed of multiple candidate cards. On the left side of the ballot card is a list of races with the candidates' names randomly ordered. On the right side of the ballot card is a box that the voter can place an "x" in for his or her candidate selection. After voters make their selections on the cards, they detach the left side of each card, which lists the races and order of candidates, from the right side, which has their selections. The voters then shred their candidates lists/left sides of their ballot cards. At this point their ballots have been anonymized so others cannot see their selections. Next the voters take the remaining parts of their ballot cards to a vote casting station. At this station they scan their ballot card selections into a system that records them and

keeps an electronic tally. After the cards are scanned in, a receipt is printed out that shows a scan of each card and the voters' marks. At this time, the original ballot cards are placed into a ballot box that is used as a paper trail in case the election needs to be audited to compare the electronic tally to the paper count. The voters keep the system-generated receipts with the images of their ballot cards that were cast. If a voter decides to verify their vote after the polls close, they navigate to the website printed on the receipt, enter the unique number associated with their ballot that is printed on the receipt, and then they are able to view the cards that were scanned into the system. This verification system was designed to illustrate that the system received the individual's specific selections without ever revealing exactly who they voted for as each race's candidates were randomly ordered on every ballot. Some of the usability problems that voters encountered while using Rice University's version of PaV included participants not reading the lengthy instructions included on the first card, resulting in failures to anonymize their ballot by ripping it in half and shredding the candidates lists. Others accidently shredded their instructions and then did not know what to do or shredded the wrong side of their ballot, requiring them to start the process over. In addition, many participants thought they cast their ballot because they confidently placed their cards in the ballot box—as they usually do when voting with a paper system, when in fact they really needed to scan the cards first so the system could record their choices and add them to the tally. For the detailed formative usability assessment and results, refer to Acemyan et al. (2015a); for photos of PaV, refer to Acemyan et al. (2014); and for details about the system's security, see papers such as Ryan (2008) and Ryan and Peacock (2005).

The third system tested by Acemyan and colleagues (2014) was Scantegrity II. Scantegrity is based on the paper bubble-ballot. To vote in an election with this system, voters used decoder pens to mark their selections. When the bubbles associated with their selections were filled in, alphanumeric codes printed in invisible ink were revealed. If voters wanted to be able to use the Scantegrity verification system, they had to hand transcribe each of these codes and the unique ballot ID onto special receipts. After the ballots and receipts were completed by the voters, the ballots were taken to a vote-casting station where they were scanned into the system and then placed into the ballot box. For the optional vote verification process, voters navigated to the election's website, entered their unique ballot ID code, and then compared one by one the codes written on their self-generated receipts to the ones displayed on the website. Errors made while using this system included voters using a regular pen or pencil to complete the ballot and then getting frustrated that they could not find their verification codes. Most participants committed transcription errors in which their ballot ID and/or one or more of their selections codes was written incorrectly; as a result, participants did not know if they or the computer system made a mistake. Moreover, some participants never cast a ballot that would be counted in the tally because they placed their completed ballot in the ballot box without ever scanning it in—likely because they either did not read the lengthy instructions on the ballot and receipt or they did not understand them, in both cases relying on typical how-to-vote procedures instead. The complete formative usability assessment can be found in Acemyan et al. (2015a); photos of assessed Scantegrity system were published in Acemyan et al. (2014); and details about the system's security properties and auditing capabilities are explained in Chaum, Carback, et al. (2008).

Overall, the usability of these tested e2e systems contrasts sharply with the usability of non-e2e systems (Byrne et al., 2007), because the e2e systems deviate substantially from the how-to-vote procedures and mental models associated with non-e2e voting methods (Acemyan et al., 2015a, 2015b). Specifically, voters expect the steps required to vote with an e2e system to be roughly the same as voting with a standard paper ballot or electronic voting machine. Instead, the e2e systems require voters to complete novel, unusual steps such as using decoding pens to reveal special codes printed in invisible ink and tearing ballots in half to separate the list of candidates from the voter selections—steps that voters do not typically associate with voting.

Besides the mismatch between mental models and the tested systems, other significant usability problems resulted in the e2e systems because the developers who are leading security

and auditing experts did not prioritize human factors and user-centered design (UCD). This is not surprising because it is outside their area of expertise, and their focus was likely on developing secure voting systems, which is a difficult problem in itself. Even though they theoretically solved the security and verification problems, the solutions resulted in system designs that were difficult or impossible to use.

Voters struggled to use the e2e systems because they had to be active participants in the security (e.g., tearing their ballot in half and then shredding part of it to anonymize it) and auditing process (e.g., creating their own labor-intensive, verification receipt)—meaning that the users are given the opportunity to perform actions that compromise the security of the systems and/or their ability to vote and verify with them. Some of the researchers realized system usability was a problem, as is evident by the intense efforts by the Scantegrity team to develop training videos and ensure that poll workers were ready to assist voters with the system during real elections (Carback et al., 2010). Essentially, they were trying to change voter behavior instead of designing a system that aligned with how voters actually behave.

It is also recognized that designing a secure and usable voting system is hard to do. Most people engage daily with secure systems that have been proven to be reasonably usable (Taohai, Phimoltares, & Cooharojananone, 2010)—such as online banking and shopping sites that use encryption. In contrast, it is difficult to make voting systems both secure and easy to use. For one, voter participation is highest among individuals over the age of 45 (File, 2014); this group tends to be less comfortable with computer use (Broady, Chan, & Caputi, 2010) compared with younger individuals who more freely engage with a variety of online systems. Second, people vote on an infrequent basis, typically once every 2 to 4 years. In contrast, many people engage in electronic banking and online shopping on a repeated basis (Rogers, Cabrera, Walker, Gilbert, & Fisk, 1996). Consequently, the benefits of learning and training are bestowed upon banking and e-commerce systems to a much greater degree than is seen in voting systems. Third, designing secure and usable voting systems is difficult because ballots are anonymous, to ensure that no one can determine for whom a person voted; in contrast, credit card transactions remain associated with the consumer who can then dispute erroneous or fraudulent charges. Despite the need for anonymity, a voting system must still be designed to prevent people from voting more than once to guard against ballot stuffing. Fourth, network and computer security experts unanimously agree that online systems such as those used for banking and shopping are not secure enough to be used for voting (Jefferson, 2016). Simply put, the security and privacy requirements of election systems are structurally different from those for e-commerce transactions. For all of these reasons, it is exceptionally challenging to build a voting system that is both secure and usable.

The Development of STAR-Vote

This led us to the question, "Is it even possible to develop a secure voting system using state-of-the art e2e security technology that is also intuitively usable by every voter?" We believe so. Accordingly, in 2011, a group of cryptographers, computer security specialists, statisticians, voting officials, and human factors researchers gathered together to develop a usable, e2e voting system. The system, called STAR-Vote (for Secure, Transparent, Auditable and Reliable), was developed using iterative, UCD, and development methods, with human factors research integrated throughout the entire process. Most notably, both usability and security were equally prioritized at every phase of the project.

From the user perspective. STAR-Vote is like many other voting systems at first glance: voters interact with a touch screen, selecting their choices, and a paper ballot is printed that the voter deposits in a locked ballot box. See Figure 1 for STAR-Vote's typical how-to-vote procedures.

Yet, STAR-Vote is in fact very secure due to features that were designed to be unnoticeable to voters, including sophisticated "end to end" (e2e) encryption (described below), multiple sources of data to support final tallies, risk-limiting auditing, and advanced software engineering (Bell et al., 2013). After a voter completes his or

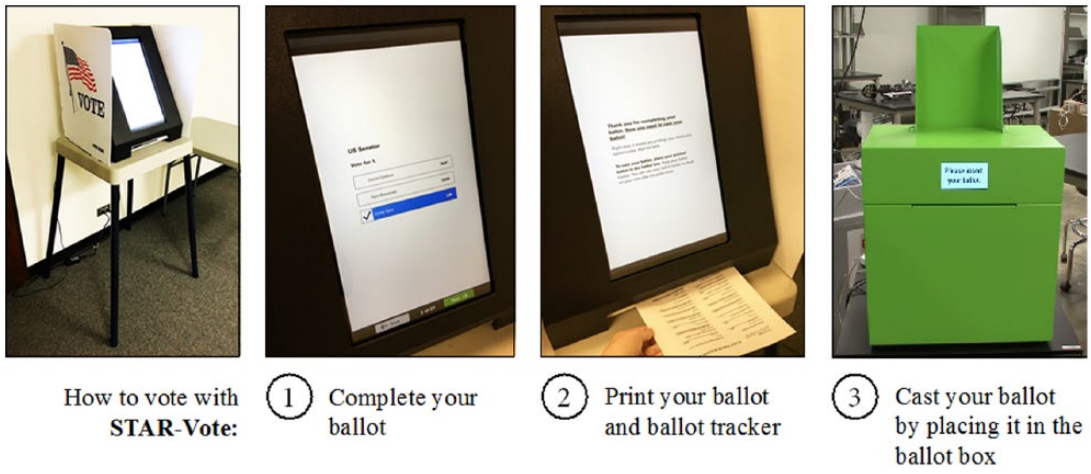

Figure 1. STAR-Vote's typical how-to-vote procedures. The security mechanisms are not apparent because they were intentionally hidden from the voters.

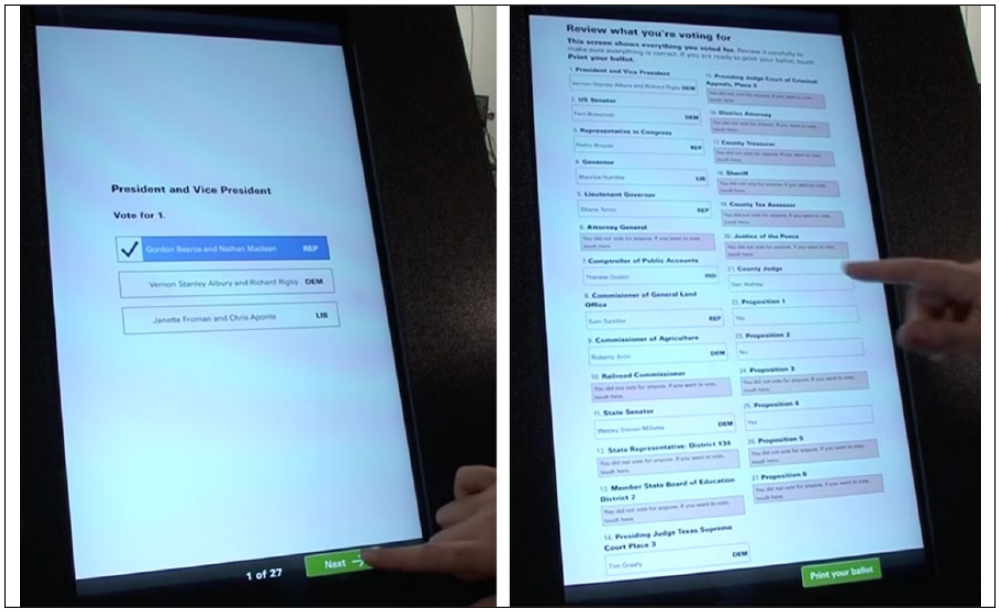

Figure 2. STAR-Vote's ballot completion interface (shown left) and ballot review screen (shown right). Simple, plain language is used (e.g., you did not select anything) and instructions are minimal and only presented at the time that they are needed by voters. Extensive instructions were not needed because the system and its interfaces were intuitive and matched typical how-to-vote procedures.

her ballot on a touch-screen tablet (Figure 2) and then prints it (Figure 3), the system automatically encrypts the voter's selections. As the voter inserts their ballot into the ballot box (Figure 4), a scanner mounted on the interior reads the ballot (thereby eliminating the separate scanning step found in PaV and Scantegrity; see Figure 5).

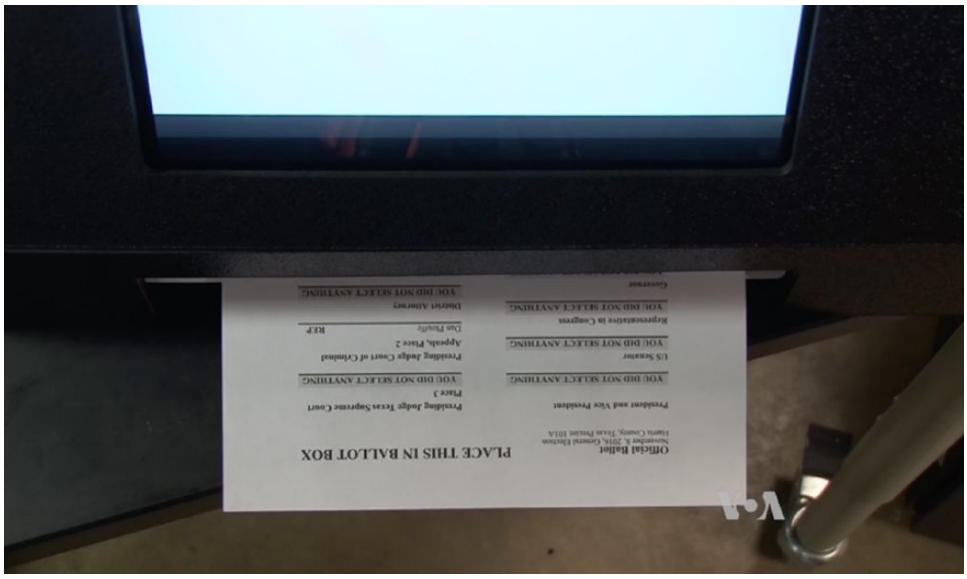

Figure 3. The ballot being printed at the voting booth. The ballot is ejected from the system directly in front of the voter so that they will notice it and take it. Voters have the opportunity to review their ballot before placing it in the ballot box.

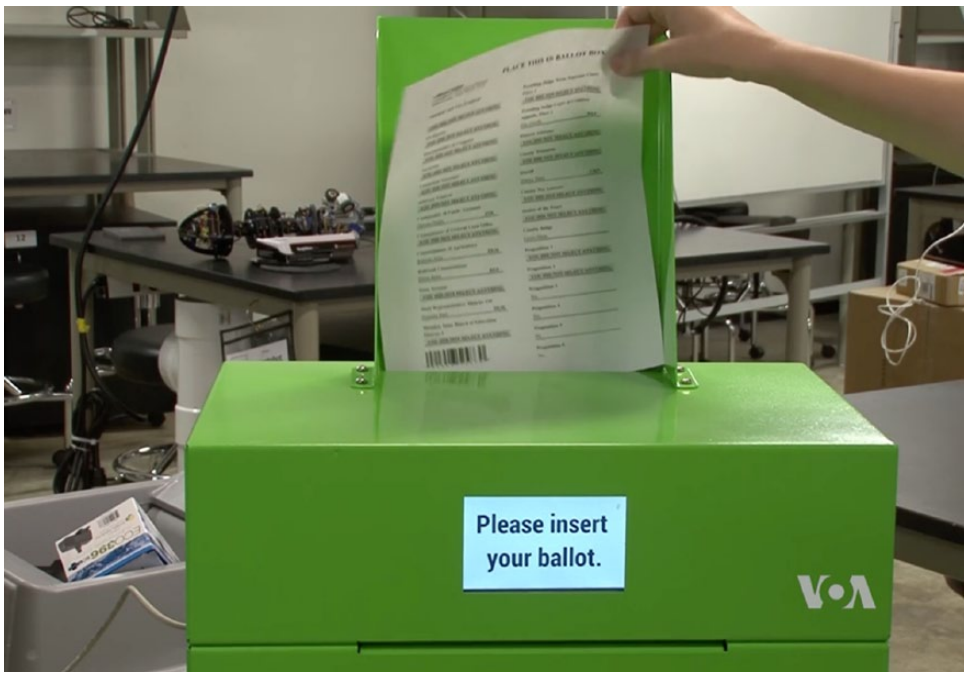

Figure 4. A voter inserts their ballot into the ballot box. Inside the ballot box is a scanner that reads the inserted material. If it is a valid ballot, the selections are recorded by the system and included in the tally; if the inserted material is an invalid ballot or a voter's receipt, it is ejected out of the front slot of the ballot box. The ballot box display will indicate if a ballot has been cast or if the ballot or material has been rejected from the system (and why). Auditory tones are also part of the design to support users who are visually impaired.

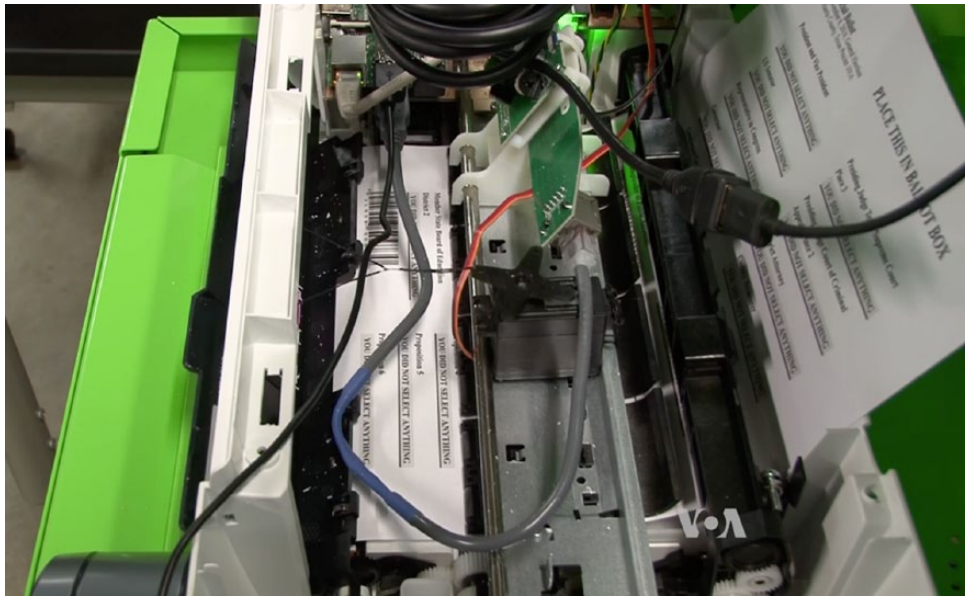

Figure 5. The ballot box's scanner and paper-diverting mechanism are concealed from voters so they do not need to worry about scanning their ballot; rather, they simply drop their ballot in the box and the system does the rest.

At that time, the box- and ballot-marking devices communicate to determine whether a valid ballot has been cast. If so, the votes for each race are recorded and added to the official tallies. If anything other than a valid ballot is placed in the box, the material is rejected by the system and ejected from the box. Because the security is (intentionally) invisible to voters and does not require their involvement, it does not alter the typical how-to-vote procedures. This reduces the potential for a mismatch between mental models for voting and the system that was seen in the previously tested e2e systems. Furthermore, the voters cannot compromise the system's security or verification process because it is completely hidden from them—making it look and act like typical voting systems without advanced security and auditing mechanisms.

STAR-Vote is also designed so that any interested person can audit the system to verify that the system is functioning as designed without ever violating anonymity. For example, (1) after polls close, voters can check that the system received their ballot by going to the election website and entering their unique ballot ID located on their personal ballot tracker (Figure 6), which was printed at the same time as their ballot. On the website they will then see whether the ballot was cast and received by the system, printed but not cast, or not found in the system (Figure 7). Individual votes for cast ballots are never revealed.

At this point in the verification process, problems can be reported. (2) The system can also be audited by election officials and/or a third party—such as the League of Women Voters—by going through a procedure to ensure that every cast vote has been recorded and counted without ever linking votes to individuals. (3) Last, additional postelection audits can be conducted by randomly sampling from the paper ballots stored in the secure ballot box to check that the paper and electronic voting records match exactly. In the event of a total failure of the electronic voting systems or a security breach, these paper ballots can be counted by hand. A complete description of the user interactions and cryptographic properties of STAR-Vote can be found in Bell et al. (2013).

STAR-Vote's e2e security model. Although this paper focuses on STAR-Vote's usability, it is worth understanding in more depth how STAR-Vote's security properties work and thus

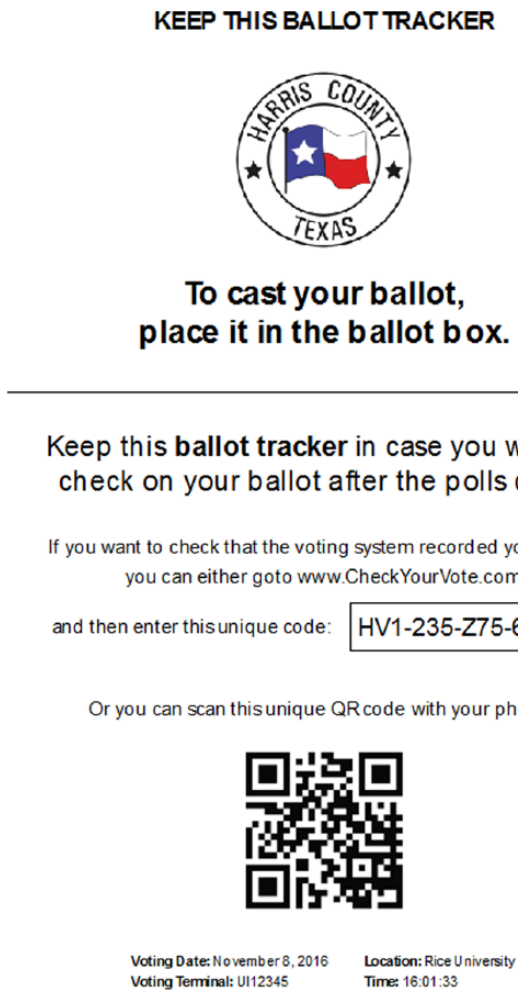

Figure 6. This ballot tracker is printed at the same time as the ballot and serves as a cover page to the ballot so that voters' selections are not revealed as they walk to the ballot box. Voters can either enter their unique code, which is chunked to aid in entry, or they can scan the QR Code for automatic entry to access the election website.

why STAR-Vote can have many desirable security features without impacting its usability. A complete discussion of STAR-Vote security can be found in Bell et al. (2013). Here we discuss three interesting properties: STAR-Vote's use of e2e cryptography, a challenge mechanism to catch the machine if it is cheating, and its use of cryptographic hash chains.

From the voter's perspective, STAR-Vote produces a printed, human-readable ballot that goes into a ballot box. Additional electronic copies are made, one on each voting terminal in the local precinct. A "public key cryptosystem" is used, meaning that every voting machine can create a ballot, but the decryption operation requires a secret key held only by election administrators, thus protecting voter privacy. Each position on the ballot corresponds to an encrypted one or zero, indicating whether the voter selected that particular candidate.

We note that it's essential that every ciphertext include a proof that it's well-formed (i.e., that it's

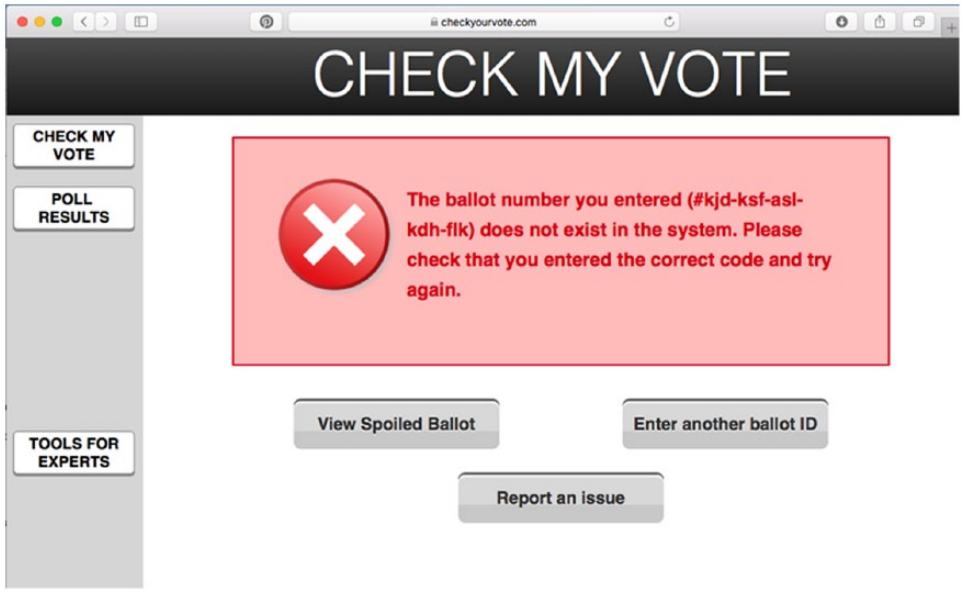

Figure 7. Voters may check the status of their ballot on the election website. Above is a screenshot of a ballot ID that does not exist in the system.

an encryption of a zero or a one, without revealing which one, but provably not being any other number). STAR-Vote uses standard Chaum-Pedersen proofs (1992) for this purpose. When the election is complete, all of the encrypted ballots are posted to a public website. Any election observer may then download these and verify the Chaum-Pedersen proofs without needing the election officials' secret cryptographic keys.

Furthermore, STAR-Vote uses a particular style of public-key cryptography called "exponential ElGamal" (1985) which has a useful "homomorphic" property. Again, without needing the secret key, any observer can use the homomorphism as follows: given the encryption of two integers $E(x)$ and $E(y)$, it's possible for any observer to derive $E(x+y)$, without needing to first decrypt and then re-encrypt. Given this, an observer can then compute an encrypted total for each ballot position. Only the election officials can produce the decryption, but they produce a proof that their decryption is consistent with the public data, which anybody can then verify (using a variant on Chaum-Pedersen for the proof, produced by the election officials). Collectively, these mechanisms produce proofs that the election was *counted as cast*, i.e., that the electronic ballots are consistent with the final totals. We refer to these as "end to end" properties because election officials can prove a direct correspondence between the voters' original ballots and the final election tally. The e2e mechanism also proves that every ballot is *well formed*—no ballot represents more than the votes that a single voter may legally cast.

One essential challenge, whenever any electronic voting system is producing an electronic record, is whether that electronic record reflects the *intent of the voter*, and thus whether they preserve ballots *cast as intended*. The software within a malicious voting machine might correctly indicate to the voter how they voted, while quietly flipping their intent in the digital record. STAR-Vote addresses this concern with a variant on a technique based on Benaloh (2006). For every vote, the machine will first *commit* to its ballot, and then a voter may *cast* the ballot or *challenge* it. If a machine manipulated the voter's intent, the challenge process would yield proof that the commitment was inconsistent, and the machine would be caught with a cryptographically signed confession of its malfeasance. Of course, a "challenged" ballot is not counted, while a "cast" ballot cannot be challenged.

Prior systems using variants on Benaloh (2006) challenges looked for ways to add one more question to the user dialog, producing opportunities for postcompletion errors. STAR-Vote encodes the cast-or-challenge question in the physical motion of the printed paper ballot, which includes a barcode tracking number. If the ballot is inserted into the ballot box, which has a barcode scanner, then the vote is cast. If the ballot is instead brought to the poll worker, it can be scanned and the challenge registered. In this way, the Benaloh challenge process fits in with existing voting procedures that accommodate mistakes a voter might make on a hand-marked paper ballot, to spoil and redo their work. Of note is the fact that both voters and voting security officials could engage in this form of checking on Election Day, further enhancing the system's security and confidence in the final vote tallies.

Lastly, STAR-Vote internally stores all of its encrypted vote records using a *hash chain*, a technique that has been popularized through cryptographic currency systems such as BitCoin. The idea is that each vote record includes a cryptographic "hash" of the previous record, which includes a "hash" of the record prior. Each record thus protects the integrity of its predecessors. If an adversary somehow substituted a different record, then the hash chain computation would fail and it would be immediately obvious that tampering had occurred. STAR-Vote adds to this a voter "receipt" that includes the hash of their encrypted vote. Printed on paper, a voter can take this home and later use it to identify the presence of their ballot among the public (encrypted) ballot records. Consequently, evidence protecting the integrity of every vote cast earlier in the day is walking out the door with each and every voter, yet voters have no ability to prove to any third-party whether they voted for or against any particular issue because they cannot decrypt their individual votes.

Development methods. In summary, STAR-Vote is an e2e cryptographically secure voting system that was designed from the outset to be usable. To maximize its usability, several strategies were used within a UCD process, which ensures that users' capabilities and limitations are taken into account and accommodated during every step of the iterative design process. This type of approach to system design helps to make certain that the voting system will be easy to use and intuitive for voters.

At the start of the UCD process, a large, interdisciplinary team composed of security researchers (including some who helped to develop the previously tested e2e systems), statistical auditors, human factors experts, and election officials attended a series of meetings to reach a consensus regarding the architecture and specifications for STAR-Vote, which had to prioritize usability and security equally. During this phase of the project, the human factors experts who had already studied voting for more than a decade relied on known HCI heuristics and findings from published human factors research to develop a list of UCD properties that should be incorporated into the system—such as presenting one race at a time; using simple, plain language that all voters could understand; providing large, visible forward and back buttons for users to navigate through the races on the ballot; enabling voters to fill out their ballot by touching their selection; telling a voter if they did not make a selection for a particular race instead of leaving it blank on their review screen; and allowing a voter to touch a race on the review screen to directly return to that race on the ballot, instead of hitting the back button numerous times. These are select examples; the list was extensive.

When system prototypes were being developed that incorporated all these elements, security and human factors researchers worked together on all phases of development. Human factors research methods for design were used to develop all user interfaces—including heuristic analysis, Pareto analysis, operational analysis, analysis of similar systems, fault trees, critical incident studies, flow analysis, task descriptions, usability assessments, and controlled experimentation (see Nemeth, 2004, for descriptions of these common techniques). The methods used varied depending on the issue that was being assessed and then fixed, as well as the design phase of the project. To give one example, early in the vote verification website design process, wireframes were used to develop each screen that a user would see, as well as the navigation flow from screen to screen. Known HF/HCI

heuristics were used to identify and fix obvious problems, and then a series of cognitive walk-throughs were performed to further refine the system. Next, the interfaces were coded so a functional prototype could be used to complete additional cognitive walk-throughs by people who were less familiar with the project. After these issues were resolved, the system went through a series of usability tests with small groups of people; as soon as a problem was identified, it was fixed, and then the next person used it. This formative method allowed us to identify the remaining issues that were not caught before and rapidly iterate. During these particular steps, the computer scientists' role was to make certain that any changes made to the front end of the system would not negatively impact the back-end of the system, which included the integrated security protocols hidden from the users.

Separate interface elements (e.g., the ballot box, the ballot completion interface, and the verification system) were developed independently but then periodically integrated into the whole system and formatively tested so that significant issues (at both the system and individual element level) could be more quickly addressed by smaller, human factors design teams focusing on a single interface that would then later also be summatively evaluated within the whole system.

Toward the end of the design process when all apparent usability problems had been resolved through numerous iterations, two larger, summative usability tests were conducted to ensure that the system could be used by a diverse sample of voters. This first study that focused on the voting system revealed a few problems that were not identified beforehand (see Study 1). Once these problems were resolved and the functionality of the prototypes further matured, the final summative assessment was run (study 2), which included the optional verification system that was ready to be tested at that time.

It took many iterations and different UCD methods to develop STAR-Vote because some problems were corner-cases that do not occur often, and sometimes problems were only unmasked when another problem was fixed. This is not surprising because the system is complex and voters are a diverse group of people, highlighting how numerous people were needed to interact with the system throughout the UCD process in order to make it truly usable. Every design decision and the numerous system iterations were informed by data collected with these methods that focused on the system users. The user-centered iterative design and development process concluded only when all apparent usability problems were addressed, as supported by the summative usability assessment results presented in this paper's Study 2. Actual voters had to be able to easily use the system to complete ballots, cast votes, and verify that the system received and counted their ballot.

Research aims. While the system was designed in a user-centered fashion, it still retains some of the characteristics, noted in the introduction, which can make e2e implementation difficult for users. Specifically, the system does not perfectly correspond to either an electronic-only voting system or a paper-only voting system. That said, the only way to be certain that STAR-Vote is usable at the end of its development cycle is to conduct summative usability assessments with real voters. Only then can researchers recognize areas where STAR-Vote needs improvement and understand how it compares to other e2e voting systems that have been previously tested (i.e., Helios, PaV, and Scantegrity).

This article presents two summative usability assessments of STAR-Vote; the second study is a replication of the first. The same experimental protocol used in numerous previously published voting studies was used, along with ISO 9241-11's suggested measurements for assessing system usability: efficiency, effectiveness, and satisfaction.

STUDY 1

Method

Participants. Participants were 30 eligible voters (i.e., 18 years or older and U.S. citizens). These participants were recruited from the Houston area through a craigslist advertisement and were paid $25 for their time. Institutional review board (IRB)–approved informed consent was obtained from each participant.

Seventeen participants (57%) were female, 12 (40%) were male, and one (3%) was transgender.

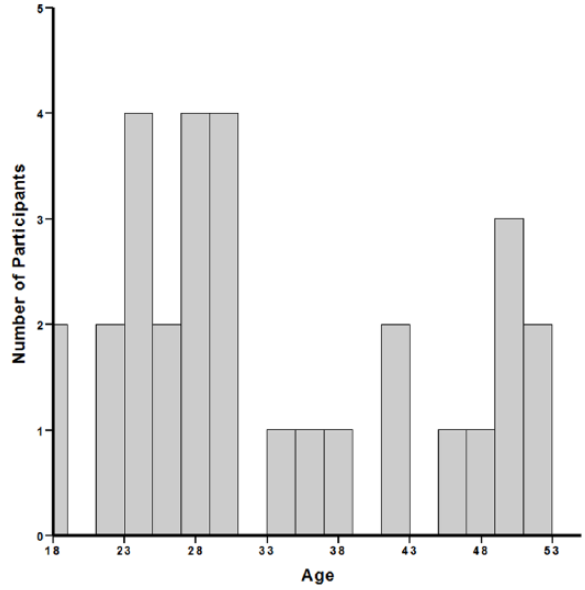

Figure 8. Age distribution for Study 1 participants.

The mean age was 33 years, with a range of 18 to 52 (see Figure 8). Seven (23%) participants were Caucasian, 11 (37%) were African American, three (10%) were Mexican American or Chicano, six (20%) were Asian American, and the remaining three (10%) were multiracial. Six (20%) participants completed a high school degree or the equivalent, eight (27%) finished some college or an associate degree, nine (30%) completed a bachelor degree or equivalent, and seven (23%) completed graduate school. When asked about their annual income, eight people (27%) indicated they earned $20,000 or less, 10 (33%) earned between $20,000 and $40,000, five (17%) earned between $40,000 and $60,000, three (10%) earned between $60,000 and $80,000, and four (13%) earned more than $80,000. In response to an item about level of computer expertise, the mean rating was 7.6 on a 10-point scale, with the range being 2–10. Many participants indicated that they had voted in a national election before, with a mean of 2.6 elections, and a range of 0 to 8. Participants voted in an average of 2.1 other types of elections (e.g., local or school board), with a range of 0 to 12. Although the sample does not directly reflect the voting population of the Houston area, it is nonetheless diverse.

Design. This study had a single condition. All participants voted with the STAR-Vote system in a mock election. So voters would know for whom they should vote, they were given a list of candidates and propositions. A mix of Republican and Democratic candidates was used, and the races, candidates, and propositions were the same as those used in more than 10 years of previous voting studies to ensure results are comparable. All candidates' names were randomly generated by a computer and were not real candidates in an election. For detailed information about the slate used in this study, refer to Acemyan et al. (2014).

Measures. ISO 9241-11 specifies that efficiency, effectiveness, and satisfaction measurements should be used to assess system usability (1998). Efficiency is the amount of time it takes a person to complete their ballot as directed and cast it. Time to vote is measured in seconds; timing began the moment participants touched the ballot marking device and ended after casting their votes. Effectiveness is a user's ability to vote without struggling and/or making errors. It is measured by noting whether a voter successfully cast a ballot and recording any mistakes made completing the ballot (i.e., did they vote for the people they were told to vote for? If

not, an error occurred). Satisfaction captures the extent that voters were satisfied using the system to complete their task. In this study, it is measured with the System Usability Scale (SUS). These measures are the same as those used in previous voting usability studies conducted at Rice University so that results can be compared across systems, especially with respect to the e2e voting systems Helios, PaV, and Scantegrity (Acemyan et al., 2014).

Materials. The STAR-Vote prototype used in this study was developed at Rice University by researchers involved since the inception of the system. System specifications were the same as those described in the STAR-Vote RFP released by Travis County to potential vendors (Travis County, 2016). The computer scientists and HF/HCI researchers iterated using HF methods in design until all apparent usability problems were resolved, as explained in this article's introduction. The ballot box accepted all material inserted into it in this version of STAR-Vote (in later iterations, a scanner inside the box read each piece of material inserted into it, then the system either accepted it by depositing the paper in the storage section of the box or rejected it by feeding it out the front ejection slot toward the user; this early prototype also did not give visual and auditory completion confirmations and warnings). In addition, the verification system had not been implemented as it was not ready for testing.

Procedures. The experimental procedure was the same as the one used in Acemyan et al. (2014, 2015a), which assessed the usability of three different types of widely accepted e2e systems that had been used in real elections. The procedures and protocol were based on the methods presented in Kortum's (2016) book on usability assessment.

After participants completed IRB-approved informed consent, they were told that they were going to be voting in a mock election. Participants were given the list of candidates for whom they were to vote. Voters began the STAR-Vote process by first checking in to receive a receipt with their precinct information. Second, they took their precinct receipt to a second check-in station to receive their voting machine PIN. Third, participants walked up to the ballot marking device, entered their PIN, and completed the ballot associated with their precinct. It was at this stage that the experimenter began taking efficiency and effectiveness measures because the check-in process is not usually included in the voting assessment per previously published voting research already cited in the article. After reviewing selections on a final screen, subjects printed their ballot and ballot tracker. Fourth, they walked up to the ballot box and cast their ballot. At the completion of this step, efficiency and effectiveness data were no longer collected. Fifth, the participants completed the SUS and study survey. Last, they were debriefed, paid, and thanked for their time.

Results

Vote casting. On average, it took participants 264 s, or 4.3 min, to vote with STAR-Vote.

The mean ballot marking error percentage was 0.23%, with a range of zero to five mistakes made on each ballot. Three ballots contained errors. Specifically, two participants (7%) made two incorrect selections, and one (3%) participant made five.

As for cast ballot rates, 29 (97%) participants successfully cast their ballots. The participant who did not complete the task handed the ballot to the experimenter instead of placing it in the ballot box and said they were finished. Nonetheless, all participants indicated that they indeed cast their ballots. The behavior and verbal response of the participant who failed to cast a vote indicates that he or she likely thought the entire vote casting procedure was completed on the ballot marking device. This error occurred despite the placement of simple instructions on both the ballot marking device's final screen and the printed ballot, which indicated in plain language using limited text in a large, bold font, "Your ballot has not been cast. To cast your ballot, you must put it in the ballot box." The ballot box was large, bright green, clearly labeled, and highly visible in the experimental room to maximize the chance that voters would notice it. Failing to complete the ballot casting process after making all selections on a ballot has been observed in previous DRE usability assessments

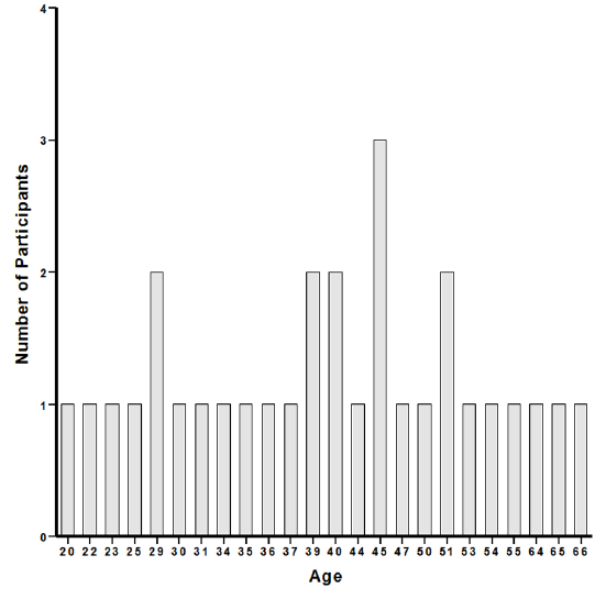

Figure 9. Age distribution for Study 2 participants.

(e.g., Everett, 2007). One possible explanation for this error might be that the voter was concentrating on completing the ballot, and once he or she finished that step, they concluded that they were done with the voting process (similar to a postcompletion error, as discussed in Everett's 2007 voting usability study). Alternately, the voter might have thought that the entire voting process took place on the voting terminal, which is possible if he or she did not read instructions or the descriptive text on the ballot completion interface. One could argue that during a real election, a poll worker would be available to redirect a voter to the ballot box—meaning the ballot would end up being cast. However, a system should never be designed to rely on poll workers (Acemyan et al., 2014, 2015a). Poll workers are not always available to help voters, receiving assistance often increases the total time to vote (which is problematic if there are long lines of people waiting to vote), and having a poll worker help a voter potentially exposes a voter's selections on their ballot.

The mean SUS score was 87.8 and the standard deviation was 13.7, with the 95% confidence interval being 82.7 to 93.0.

STUDY 2

Method

With the following exceptions, the methods for Study 2 were the same as those used in Study 1:

Participants. Thirty people who were eligible U.S. voters participated in the second study. Sixteen (53%) participants were female, 13 (43%) were male, and one (3%) was transgender. The mean age was 41.47 years, with a range of 20–66 (see Figure 9). Seventeen (57%) participants were Caucasian, eight (27%) were African American, two (7%) were Mexican American or Chicano, and the remainder were other Hispanic or Latino (1 / 3%), Nigerian (1 / 3%), American Indian (1 / 3%), and other (1 / 3%). Three (10%) participants completed a high school degree or the equivalent, 12 (40%) finished some college or an associate degree, 10 (33%) completed a bachelor degree or equivalent, and five (17%) completed graduate school. When asked about their annual income, nine people (30%) indicated they earned $20,000 or less, nine (30%) earned between $20,000 and $40,000, six (20%) earned between $40,000 and $60,000, three (10%) earned between $60,000 and $80,000, and three (10%) earned more than

$80,000. The mean rating for self-reported level of computer expertise was 7.7 on a 10-point scale, with the range being 3–10. Many participants indicated that they had voted in a national election before, with a mean of 4.8 elections, and a range of 0 to 18. Participants voted in a mean of 8.2 other types of elections, with a range of 0 to 30 elections. Like Study 1, this sample is not perfectly representative of all Houston-area voters, but it is diverse.

Materials. The STAR-Vote beta-prototype used in this study included a fully functional ballot box, which internally scanned ballots and rejected invalid materials through a front ejection slot. The ballot box also gave users both visual and auditory confirmation messages and warnings at the time they were needed. In this version of STAR-Vote, the vote verification system was also implemented. After voting, participants kept their ballot tracker so they could later verify that the system received their ballot. They then navigated to the election website www.checkyourvote.com and entered in their unique ballot ID on the main page. After submitting their ballot ID, the system would report if the ballot had been recorded by the system. On the same screen, the verification system offered the options of entering another ballot ID or reporting an issue.

Design. The usability of the voting system (i.e., the components used to complete a ballot and cast it) was assessed independently of the optional vote verification system. This practice aligns with the previous protocol used to assess the usability of e2e voting systems (Acemyan, 2014), which is necessary to compare results across systems.

Procedures. In Study 2, voters did not complete the two-part STAR-Vote check-in process. Instead, participants immediately started voting. The check-in process was dropped because it did not impact the collected measurements such as vote timing, which only included the time it took to complete a ballot and cast it. A potential problem with this change in protocol is that it could potentially impact SUS scores if voters did not like the previously included check-in process or found it to be difficult.

Results

Vote casting. As can be seen in Figure 10, the mean voting time was 272 s, or about 4.5 min, with a standard deviation of 80.5 s. When comparing time on task across the two studies, the effect was not reliable, t (57.8) = 0.41, p = .681, Cohen's d = 0.11. Any differences in voting times across the two versions of STAR-Vote are trivial.

Regarding effectiveness, participants did not make errors when completing their ballot, so the mean ballot marking error rate was 0% (see Figure 11). There is not a reliable difference in error rates across the two studies, t (58) = 1.37, p = .177, Cohen's d = 0.35. The rate of successfully casting the entire ballot was 93% (see Figure 12); two (7%) of the 30 participants did not successfully cast their ballots. One participant indicated that they were done voting when they reached the review screen. When the experimenter asked if they cast their ballot, the participant replied, "yes." The second participant printed their ballot and then handed it to the experimenter instead of placing it in the ballot box. When asked if they cast a ballot that would be counted in the election, the participant replied, "yes." This is the same type of error observed in Study 1 and Everett's (2007) project that examined the usability of electronic voting machines (i.e., DREs, or direct recording electronic voting systems). Future research needs to focus on understanding and solving this fleeing voter problem because it occurs when voting with numerous types of electronic and online voting systems, even those that do not involve paper. It also complicates real elections, because poll officials do not know if voters intended to abandon their ballot and not cast it, or if they wanted to cast the ballot and did not realize that they failed to do so.

The mean satisfaction rating for vote casting was 89.2, with a standard deviation of 16.9, on the SUS scale (see Figure 13). The range was 17.5 to 100. Again, there was not a reliable difference in satisfaction ratings across the two studies, t (58) = 0.34, p = .738, Cohen's d = 0.09.

Vote verification. The mean vote verification time was 112.3 s, or a little less than 2 min, with a standard deviation of 53.3 s (see Figure 14). Of the 28 participants who successfully cast their ballot, 100% verified online that their ballot was cast and counted by the STAR-Vote system (see Figure 15). As can be seen in Figure 16, the

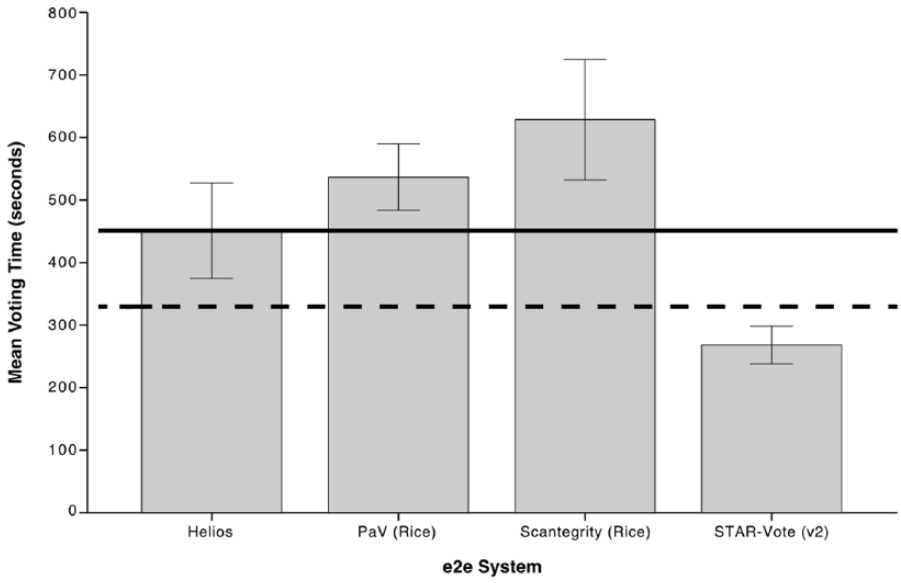

Figure 10. Mean vote casting time, in seconds, as a function of e2e voting system, with error bars representing 95% confidence intervals. Results for Helios, Prêt à Voter (PaV), and Scantegrity were reported in Acemyan et al., 2014; results for the paper bubble ballot—shown with a dotted line—were reported in Byrne, Greene, and Everett, 2007; and results for the electronic voting machine (DRE)—represented by the solid line—were reported in Everett, Greene, Byrne, and Wallach, 2008.

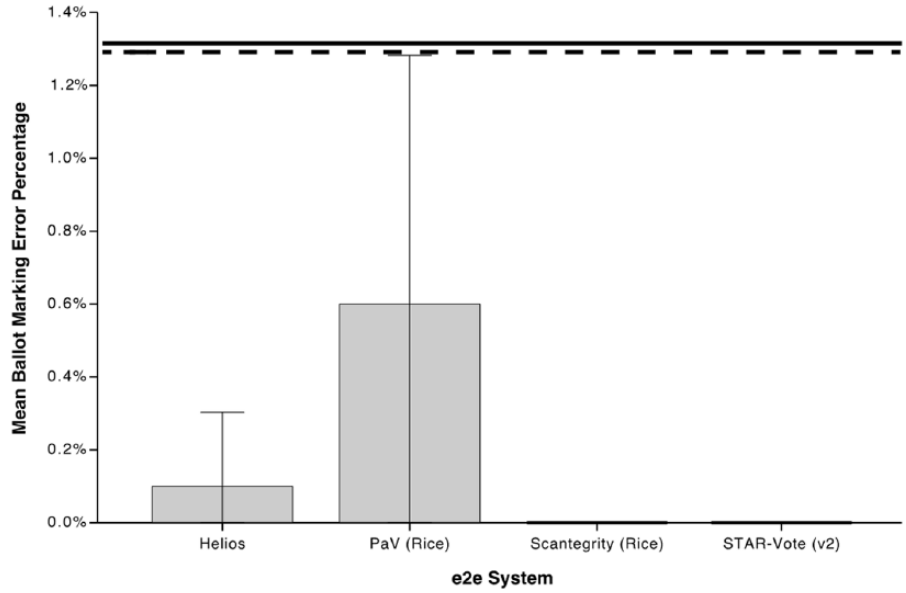

Figure 11. Mean ballot marking error percentage as a function of e2e voting system, with error bars representing 95% confidence intervals. Results for Helios, PaV, and Scantegrity were reported in Acemyan et al., 2014; results for the paper bubble ballot—shown with a dotted line—were reported in Byrne et al., 2007; and results for the electronic voting machine (DRE)—represented by the solid line—were reported in Everett et al., 2008.

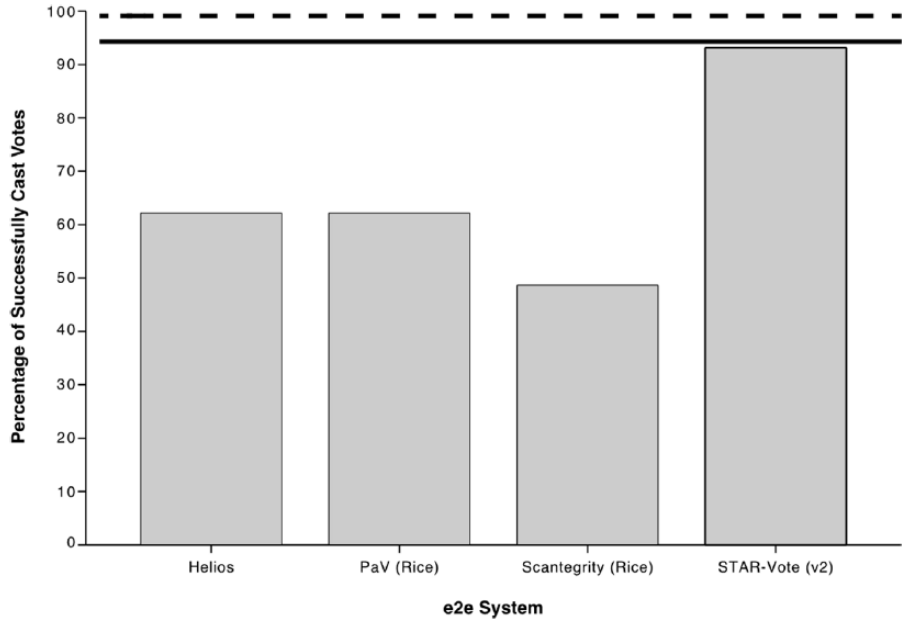

Figure 12. Percentage of successfully cast votes as a function of e2e voting system. Results for Helios, PaV, and Scantegrity were reported in Acemyan et al., 2014; results for the paper bubble ballot—shown with a dotted line—were reported in Byrne et al., 2007; and results for the electronic voting machine (DRE)—represented by the solid line—were reported in Everett et al., 2008.

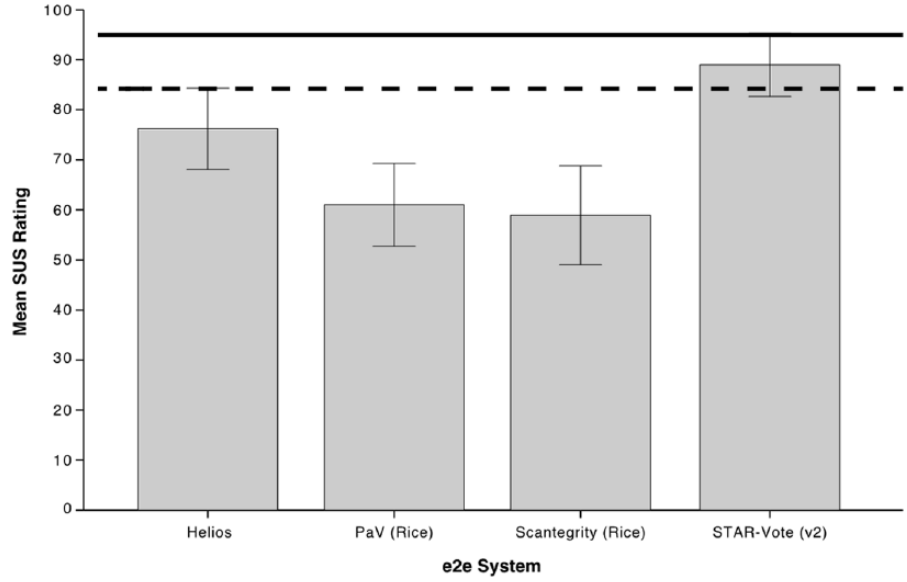

Figure 13. Mean System Usability Scale (SUS) score as a function of e2e voting system, with error bars representing 95% confidence intervals. Results for Helios, PaV, and Scantegrity were reported in Acemyan et al., 2014; results for the paper bubble ballot—shown with a dotted line—were reported in Byrne et al., 2007; and results for the electronic voting machine (DRE)—represented by the solid line—were reported in Everett et al., 2008.

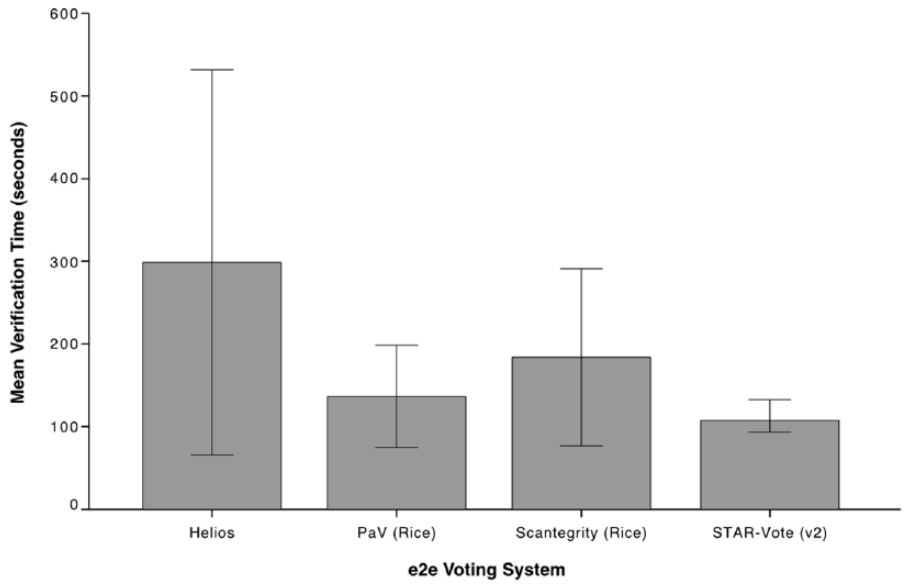

Figure 14. Mean vote verification time, in seconds, as a function of e2e vote verification system, with error bars representing 95% confidence intervals; results for Helios, PaV, and Scantegrity were reported in Acemyan et al., 2014.

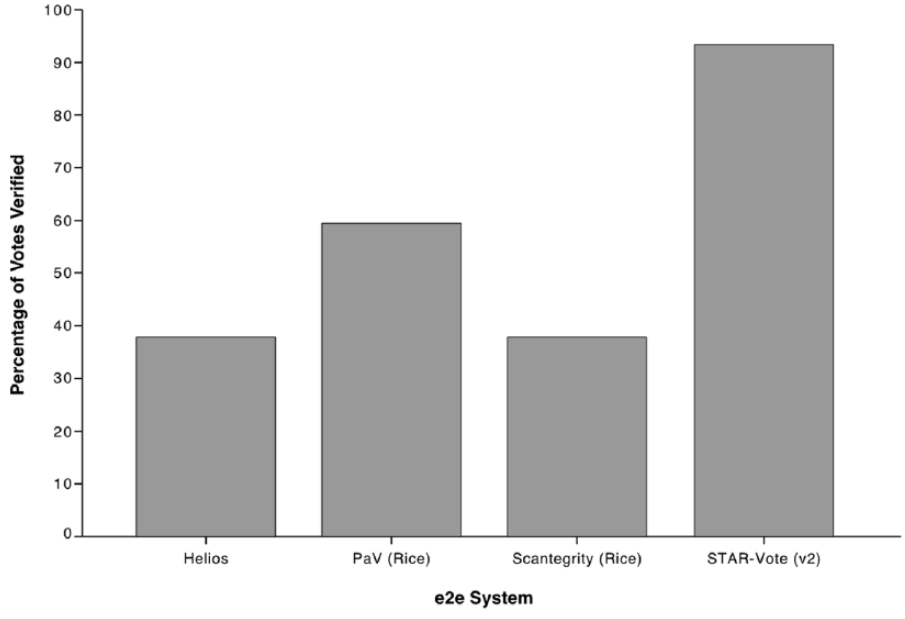

Figure 15. Percentage of verified votes as a function of e2e vote verification system; results for Helios, PaV, and Scantegrity were reported in Acemyan et al., 2014.

mean SUS rating for the vote verification system was 87, with a standard deviation of 19.3 and a range of 10 to 100.

A visual comparison of STAR-Vote to previously tested e2e voting systems. Our laboratory has conducted previous studies where the

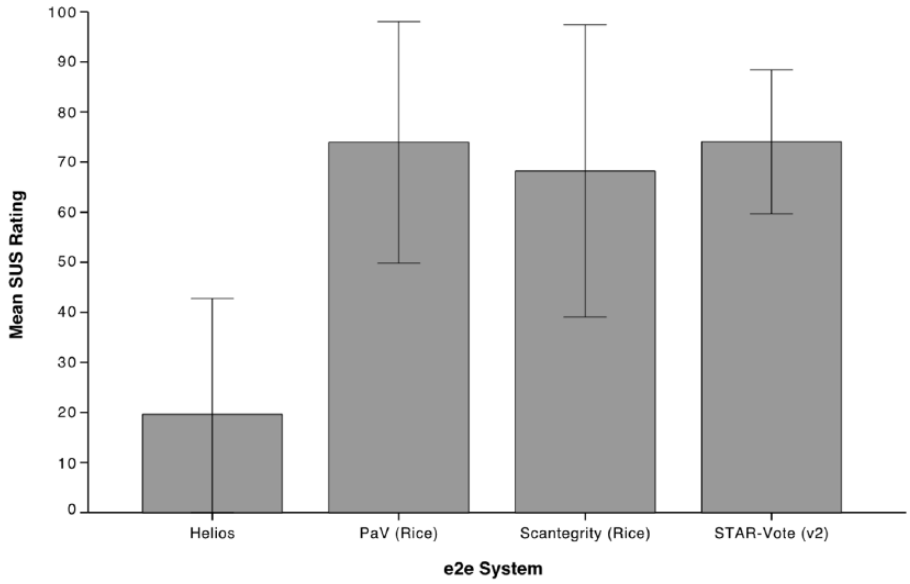

Figure 16. Mean System Usability Scale (SUS) score as a function of e2e vote verification system, with error bars representing 95% confidence intervals; results for Helios, PaV, and Scantegrity were reported in Acemyan et al., 2014.

usability of a number of different voting systems was examined. Although it is unorthodox to compare across studies, it is highly informative to understand, at the level of descriptive performance statistics, how STAR-Vote visually compares to these other voting systems. As can be seen across Figures 10–16, voters' performance across measures visually appears to be much better on STAR-Vote than the e2e voting systems that were previously tested in the authors' labs using the exact same methods as this paper's (Acemyan et al., 2014; 2015a). This is not surprising since STAR-Vote's user interface was developed from its inception using a UCD process, which included integrated usability testing. Moreover, the competing demands of security, statistical auditability, and usability were simultaneously considered throughout the system design and development process so that the system could be both highly usable and highly secure.

DISCUSSION

Although recent systems with e2e cryptography such as Helios and Prêt à Voter appear to have good security properties, they are weak from a usability perspective. In general, the usability problems stem from additional steps voters must take that do not match their mental models for voting systems. Moreover, voters do not fully understand the need for these extra steps and secure voting systems (Acemyan et al., 2014, 2015a, 2015b). This resulted in critical failures such as users failing to cast their ballots at all, and users being unable to verify their ballots. STAR-Vote successfully addressed both of these issues. Furthermore, the error rate in completing the ballot with STAR-Vote was zero, the time to complete the ballot was lower, and satisfaction scores were higher than with other e2e systems per a visual inspection of the Figures. These results not only appeared to be better than other e2e systems, but also superior to paper ballots and previous non-e2e electronic voting systems. This suggests that STAR-Vote is a highly usable system when compared across studies to previously tested voting systems.

The usability of STAR-Vote is a significant accomplishment for multiple reasons. First, advances in systems that protect election integrity are critical. Election integrity is a complex problem. Ensuring that the final election results reflect the will of the voters requires protection of the voting system at many levels. Historically,

the focus has been on the physical security of the ballots themselves, e.g., locking the ballot box. Now that many voting systems are computerized, protections such as cryptography are almost certainly necessary to prevent newer, more sophisticated methods of tampering with ballots. However, if what is on the ballots does not accurately reflect the will of the voter, there is no method of securing the ballots that will ensure that the election outcome is legitimate. Voter intent must be correctly captured by the voting system, and voters must be able to successfully cast their ballots.

Second, making STAR-Vote both usable and secure serves as yet another illustration of key human factors principles: usability must be considered early in the design phase and iterated through the design and development process using a UCD process. System usability and users' needs cannot become a consideration only after other aspects of the system have been designed. It is likely that previous systems failed to perform well on the usability front exactly because human factors were not represented early and often in the design and development process for these systems. In contrast, human factors were given a seat at the table in the initial design meetings that generated STAR-Vote, and many design ideas were vetoed due to anticipated poor usability. The design goal from the outset was to make sure that the voter did not have to adopt a new mental model of voting and could vote using typical how-to-vote procedures. We thought hard about how to make the end-to-end properties invisible to the voter, producing minimal additional overhead. Although this was the intent, intent alone does not ensure good human factors outcomes. Thus, we further iterated on the user interface of the STAR-Vote prototype with formative testing, and then subjected it to the summative tests reported here.

Ironically, security experts have exactly the same issue with not being included throughout the design and development process: security must be a consideration from the outset and cannot just be tacked on at the end of the design. Therefore, the design of voting systems must include experts on security, usability, and election administration from the beginning of the UCD process. This is what will have to happen going forward if jurisdictions want voting systems that are both usable and secure, but also practical from an election administration standpoint.

However positive these results presented in this article are, much work remains to be done. For example, the carefully-worded instructions used in STAR-Vote will have to be re-evaluated in different languages that must be supported in elections. Accessibility for voters with disabilities is a challenge for any voting system, and it is still unclear the extent to which STAR-Vote, and in particular its e2e components, meet this challenge. Group differences have been the focus of voting usability research (and most human factors research), therefore individual differences need to be studied to ensure every voter can use the system easily and independently. This relates to the limitations of this article's participant samples. Although the participants were a diverse group of real voters, they were not truly representative of *all* voters—especially with concern to age and education. This limitation, along with using a more diverse set of recruitment methods that do not rely on people (or their friends) to have access to computers and the internet, can be addressed in future research.

We also do not know whether voters believe STAR-Vote is more trustworthy and secure than more vulnerable voting systems such as those used in current elections, so future research could explore this question. Furthermore, usability of STAR-Vote from the perspective of the poll worker has also not yet been examined. Despite these unresolved issues, we believe the current STAR-Vote beta-prototype is the most usable e2e encrypted voting system to date and represents the first time that an e2e system has empirically demonstrated acceptable usability for both vote casting and vote verification.

In conclusion, STAR-Vote contributes not just to voting system research but also to the field of human factors by demonstrating that system security and system usability are not necessarily competing design goals. Although it is difficult to achieve both strong security and usability, not trading off one for the other, STAR-Vote illustrates how interdisciplinary teams that rely on an iterative user-centered design approach can accomplish this goal. Both usabil-

ity and security professionals constantly advocate the need to be involved early in the design process; neither can be adequately achieved by being an add-on at the end of development. By including both from the outset, STAR-Vote is a usable, e2e system that can be used in real, large-scale elections at a time when there is an increased focus on election tampering by foreign adversaries.

ACKNOWLEDGMENTS

We would like to thank Morgan Lewis and Levi Saucedo for their assistance in collecting the data. Matt Bernhard, Matt Kindy, Clayton Drazner, Brian Tran, Grant Belton, Sarah Brooks, and Jerry Lu participated in the development of the research prototypes. This work was supported in part by NSF grant CNS-1409401.

KEY POINTS

- STAR-Vote is both a highly usable and secure system when compared with previously tested voting systems.
- System usability cannot become a consideration only after other aspects of the system, such as security, have been designed.
- Although security and usability are often believed to be antagonistic to one another, it is entirely possible to develop a system that is both highly secure and highly usable if a user-centered design process is used.

REFERENCES

Acemyan, C. Z., Kortum, P., Byrne, M. D., & Wallach, D. (2014). Usability of voter verifiable end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II. *Journal of Election Technology and Systems*, *2*(3), 26–56.

Acemyan, C. Z., Kortum, P., Byrne, M. D., & Wallach, D. S. (2015a). From error to error: Why voters could not cast a ballot and verify their vote with Helios, Prêt à Voter, and Scantegrity II. *Journal of Election Technology and Systems*, *3*(2), 1–25.

Acemyan, C. Z., Kortum, P., Byrne, M. D., & Wallach, D. S. (2015b). Users' mental models for three end-to-end voting systems: Helios, Prêt à Voter, and Scantegrity II. *Human Aspects of Information Security, Privacy, and Trust: Lecture Notes in Computer Science*, *9190*, 463–474.

Adida, B. (2008). Helios: Web-based open-audit voting. *USENIX Security Symposium*, *17*, 335–348.

Balfanz, D., Durfee, G., Grinter, R. E., & Smetters, D. K. (2004). In search of usable security: Five lessons from the field. *IEEE Security & Privacy*, *5*, 19-24.

Balzarotti, D., Banks, G., Cova, M., Felmetsger, V., Kemmerer, R. A., Robertson, W., Valeur, F., & Vigna, G. (2010). An experience in testing the security of real-world electronic voting systems. *IEEE Transactions on Software Engineering*, *36*(4), 453–473.

Bederson, B. B., Lee, B., Sherman, R. M., Herrnson, P. S., & Niemi, R. G. (2003). Electronic voting system usability issues. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 145–152). New York, NY: ACM.

Bell, S., Benaloh, J., Byrne, M. D., DeBeauvoir, D., Eakin, B., Fisher, G., . . . Pereira, O. (2013). STAR-Vote: A secure, transparent, auditable, and reliable voting system. *USENIX Journal of Election Technology and Systems (JETS)*, *1*(1), 8.

Benaloh, J. (2006). Simple verifiable elections. *USENIX /Accurate Electronic Voting Technology Workshop*, *6*, 5.

Bishop, M. (2007). *Overview of red team reports. Top to bottom review of electronic voting machines*. Office of the Secretary of State of California, Sacramento, CA.

Broady, T., Chan, A., & Caputi, P. (2010). Comparison of older and younger adults' attitudes towards and abilities with computers: Implications for training and learning. *British Journal of Educational Technology*, *41*(3), 473–485.

Byrne, M. D., Greene, K. K., & Everett, S. P. (2007). Usability of voting systems: Baseline data for paper, punch cards, and lever machines. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 171–180). New York, NY: ACM.

Calandrino, J. A., Feldman, A. J., Halderman, J. A., Wagner, D., Yu, H., & Zeller, W. P. (2007). Source code review of the Diebold voting system. *University of California, Berkeley under contract to the California Secretary of State*.

Campbell, B. A., & Byrne, M. D. (2009a). Now do voters notice review screen anomalies? A look at voting system usability. In *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*. Berkeley, CA: USENIX Association.

Campbell, B. A., & Byrne, M. D. (2009b). Straight-party voting: What do voters think? *IEEE Transactions on Information Forensics and Security*, *4*(4), 718–728.

Carback, R., Chaum, D., Clark, J., Conway, J., Essex, A., Herrnson, P. S., . . . Sherman, A. T. (2010). Scantegrity II municipal election at Takoma Park: The first E2E binding governmental election with ballot privacy. Retrieved from http://static.usenix.org/legacy/events/sec10/tech/full_papers/Carback.pdf

Cazier, J. A., & Medlin, B. D. (2006). Password security: An empirical investigation into e-commerce passwords and their crack times. *Information Systems Security*, *15*(6), 45–55.

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. *EVT*, *8*, 1–13.

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

Chaum, D., & Pederson, T. (1992). Wallet databases with observers. *Advances in Cryptology – Crypto '92,* Volume 740 of *Lecture Notes in Computer Science*, Springer-Verlag, 89–105.

Conrad, F. G., Bederson, B. B., Lewis, B., Peytcheva, E., Traugott, M. W., Hanmer, M. J., . . . Niemi, R. G. (2009). Electronic voting eliminates hanging chads but introduces new usability challenges. *International Journal of Human-Computer Studies*, *67*(1), 111–124.

Cranor, L. F., & Garfinkel, S. (2005). *Security and usability: Designing secure systems that people can use*. Sebastopol: O'Reilly Media, Inc.

ElGamal, T. (1985). A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory 31*(4), 469–472.

Entous, A., & Nakashema, E. (2016). FBI in agreement with CIA that Russia aimed to help Trump win White House. *Washington Post*. Retrieved from https://www.washingtonpost.com/politics/clinton-blames-putins-personal-grudge-against-her-for-election-interference/2016/12/16/12f36250-c3be-11e6-8422-eac61c0ef74d_story.html?utm_term=.47621777212a.

Epstein, J. (2015). Weakness in depth: A voting machine's demise. *IEEE Security & Privacy*, *13*(3), 55–58.

Everett, S. P. (2007). *The usability of electronic voting machines and how votes can be changed without detection* (Unpublished doctoral dissertation). Rice University, Houston, TX.

Everett, S. P., Greene, K., Byrne, M., & Wallach, D. (2008). Electronic voting machines versus traditional methods: Improved preference, similar performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 883–892.

Feldman, A. J., Halderman, J. A., & Felten, E. W. (2007). Security analysis of the Diebold AccuVote-TS voting machine. In *Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 07)*. Retrieved from https://www.usenix.org/legacy/event/evt07/tech/full_papers/feldman/feldman_html/

File, T. (2014, April). *Young-adult voting: An analysis of presidential elections, 1964–2012*. Washington, DC: U.S. Census Bureau.

Garfinkel, S., & Lipford, H. R. (2014). Usable security: History, themes, and challenges. *Synthesis Lectures on Information Security, Privacy, and Trust*, *5*(2), 1–124.

Halderman, J. A., & Teague, V. (2015). The New South Wales iVote system: Security failures and verification flaws in a live online election. *International Conference on E-Voting and Identity*, 35–53. doi:https://doi.org/10.1007/978-3-319-22270-7_3

Help America Vote Act of 2002, Pub.L. No. 107-252 (2002).

ISO. (1998). 9241-11. *Ergonomic requirements for office work with visual display terminals (VDTs)—Part 11: Guidance on usability*. Geneva: The International Organization for Standardization.

Jefferson, D. (2016). If I can shop and bank online, why can't I vote online? *Verified Voting: Safeguarding Elections in the Digital Age*. Retrieved from https://www.verifiedvoting.org/resources/internet-voting/vote-online/

Kortum, P. (2016). *Usability assessment: How to measure the usability of products, services, and systems*. Santa Monica, CA: Human Factors and Ergonomics Society.

Langone, A. (2018, August 14). An 11-year-old hacked into a U.S. voting system replica in 10 minutes this weekend. *Time*. Retrieved from http://time.com/5366171/11-year-old-hacked-into-us-voting-system-10-minutes/

Laskowski, S. J., Autry, M., Cugini, J., Killam, W., & Yen, J. (2004). Improving the usability and accessibility of voting systems and products. *NIST Special Publication*, *500*, 256.

Nemeth, C. (2004). *Human factors methods for design*. Boca Raton, FL: CRC Press.

Norman, D. A. (2010). The way I see it: When security gets in the way. *Interactions*, *16*(6), 60–63.

Paul, N., Evans, D., Rubin, A., & Wallach, D. (2003). Authentication for remote voting. In *Workshop on Human-Computer Interaction and Security Systems*, Fort Lauderdale, FL.

Payne, B. D., & Edwards, W. K. (2008). A brief introduction to usable security. *IEEE Internet Computing*, *12*(3), 13–21.

Proebstel, E., Riddle, S., Hsu, F., Cummins, J., Oakley, F., Stanionis, T., & Bishop, M. (2007). An analysis of the Hart Intercivic DAU eSlate. In *2007 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 07)*.

Rogers, W. A., Cabrera, E. F., Walker, N., Gilbert, D. K., & Fisk, A. D. (1996). A survey of automatic teller machine usage across the adult life span. *Human Factors*, *38*(1), 156–166.

Ryan, P. (2008). Prêt à Voter with Paillier encryption. *Mathematical and Computer Modeling*, *48*, (9–10), 1646–1662.

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., & Peacock, T. (2005). *Prêt à Voter: A system perspective*. University of Newcastle upon Tyne. Retrieved from https://www.researchgate.net/profile/Peter_Ryan7/publication/277296393_Pret_a_voter_a_systems_perspective/links/55c0c57b08aec0e5f447ad8e.pdf

Sandler, D., Derr, K., & Wallach, D. S. (2008, July). VoteBox: A tamper-evident, verifiable electronic voting system. In *USENIX Security Symposium*, 4(0), 87.

Sasse, M. A., Brostoff, S., & Weirich, D. (2001). Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security. *BT Technology Journal*, *19*(3), 122–131.

Scarfone, K., & Souppaya, M. (2009). *Guide to enterprise password management (draft)*. NIST Special Publication, 800, 118.

Stanton, B. C., & Greene, K. K. (2014). Character strings, memory and passwords: What a recall study can tell us. In *Human Aspects of Information Security, Privacy, and Trust* (pp. 195–206). Berlin: Springer International Publishing.

Stanton, J. M., Stam, K. R., Mastrangelo, P., & Jolton, J. (2005). Analysis of end user security behaviors. *Computers & Security*, *24*(2), 124–133.

Summers, W. C., & Bosworth, E. (2004). Password policy: The good, the bad, and the ugly. In *Proceedings of the Winter International Symposium on Information and Communication Technologies* (pp. 1–6). Trinity College: Dublin.

Taohai, K., Phimoltares, S., & Cooharojananone, N. (2010). Usability comparisons of seven main functions for automated teller machine (ATM) banking service of five banks in Thailand. In *2010 International Conference on Computational Science and Its Applications (ICCSA)* (pp. 176–182). Fukuoka: IEEE.

Tognazzini, B. (2005). Designing for usability. In L. F. Cranor & S. Garfinkel, (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 31–46). Sebastopol, CA: O'Reilly.

Travis County, Texas. (2016). *Request for proposals (RFP) STAR-Vote: A new voting system* (RFP # P1609-008-LC). Austin, TX: Travis County Purchasing Office.

Wand, J. N., Shotts, K. W., Sekhon, J. S., Mebane Jr, W. R., Herron, M. C., & Brady, H. E. (2001). The butterfly did it: The aberrant vote for Buchanan in Palm Beach County, Florida. *American Political Science Review*, 793–810.

Claudia Ziegler Acemyan is an adjunct assistant professor in the psychological sciences department at Rice University in Houston, Texas. She is also a senior human factors engineer with KBRwyle at NASA Johnson Space Center. Acemyan completed her PhD in human factors/HCI psychology at Rice University in 2014.

Philip Kortum is an associate professor in the department of psychological sciences at Rice University. Kortum received his PhD in biomedical engineering from University of Texas at Austin in 1994.

Michael D. Byrne is a professor in the departments of psychological sciences and computer science at Rice University. Byrne completed his PhD in experimental psychology at Georgia Institute of Technology in 1996.

Dan S. Wallach is a professor in the departments of computer science and electrical and computer engineering and a Rice Scholar at the Baker Institute for Public Policy at Rice University. Wallach earned his PhD in computer science from Princeton University in 1999.

Date received: June 5, 2017
Date accepted: October 14, 2018