Chapter 14

STAR-Vote: A Secure, Transparent, Auditable and Reliable Voting System

Susan Bell
Office of the Travis County Clerk

Josh Benaloh
Microsoft Research

Michael D. Byrne
Rice University

Dana DeBeauvoir
Office of the Travis County Clerk

Bryce Eakin
Independent Researcher

Gail Fisher
Office of the Travis County Clerk

Philip Kortum

Rice University

Neal McBurnett

ElectionAudits

Julian Montoya

Office of the Travis County Clerk

Michelle Parker

Office of the Travis County Clerk

Olivier Pereira

Université catholique de Louvain

Philip B. Stark

University of California, Berkeley

Dan S. Wallach

Rice University

Michael Winn

Office of the Travis County Clerk

CONTENTS

(An earlier version of this chapter, with the same authors and title, was published in the *USENIX Journal of Election Technologies and Systems (JETS)*, volume 1, number 1, August 2013.)

14.1 Introduction

A decade ago, DRE voting systems promised to improve many aspects of voting. By having a computer mediating the user's voting experience, they could ostensibly improve usability through summary screens and a variety of accessibility features including enlarged text, audio output, and specialized input devices. They also promised to improve the life of the election administrator, yielding quick, accurate tallies without any of the ambiguities that come along with hand-marked paper ballots. And, of course, they were promised to be secure and reliable, tested and certified. In practice, DRE systems had problems in all these areas.

Many current DRE voting systems experienced their biggest sales volume after the failures of punch card voting systems in Florida in the 2000 presidential election. The subsequent Help America Vote Act provided a one-time injection of funds that made these purchases possible. Now, more than a decade later, these machines are near the end of their service lifetimes.

In 2012, the Travis County election administration, having used Hart InterCivic's eSlate DRE system for over a decade, concluded that no system on the market—DRE or optical scan—met their future needs. They prefer to avoid hand-marked paper ballots because they open the door to ambiguous voter intent, a source of frustration in their previous centrally-tabulated optical scan system. They didn't want to go back.

Travis County's needs and preferences impose several significant constraints on the design of STAR-Vote:

DRE-style UI Hand-marked ballots are not to be used, for the reason above. DRE-style systems were also preferred for their ability to offer facilities for voters with disabilities.

Printed paper ballot summaries While the DRE-style UI was desired for entering selections, printed ballots were desired for their security benefits, verifiability by voters and redundancy against failures in the electronic system. In order to save on paper and paper management, the county wished to only print a list of each voter's selections, analogous to the summary screens on many current-generation DRE systems.

All-day battery life Power failures happen. Current-generation DRE systems have batteries that can last for hours. The new system must also be able to operate for hours without external power.

Early voting and election-day vote centers Travis County supports two weeks of early voting, where any voter may vote in any of more than 20 locations. Also, on election day, any voter may go to any local polling place. Our county's voters informally report their appreciation of these benefits.

COTS hardware Current DRE systems are surprisingly expensive. Travis County wants to use commercially available, off-the-shelf equipment, whenever possible, to reduce costs and shorten upgrade cycles. That is, "office equipment" rather than "election equipment" should be used where possible.

Long ballots While voters in many countries only select a candidate for member of parliament, in the U.S., voters regularly face 100 or more contests for federal, state and regional offices; judges; propositions; and constitutional amendments. STAR-Vote must support very long ballots as well as long lists of contestants in each race.

These constraints interact in surprising ways. Even if the county did not have a strong preference for a DRE-like UI, pre-printed paper ballots are inefficient for vote centers, which may need to support hundreds or thousands of distinct ballot styles. Likewise, the requirement to run all day on battery backup eliminates the possibility of using laser printers for ballot-on-demand printing, which consume far too much power.[1] We note that counties that face fewer constraints may choose to adopt quite different architectures. For example, a county without election-day vote centers might instead use pre-printed ballots and electronic ballot marking devices.

These constraints likewise eliminate prior-fielded E2E systems like Scantegrity [133, 144], and Prêt à Voter [502, 125], which rely on hand-marked paper, and Helios [46, 44], which is meant for use in web browsers, not traditional polling locations. Wombat [80] has a paper trail, but it's only designed for single-issue ballots. VoteBox [510] has a DRE-like interface, but it's an entirely paperless system. Instead, to satisfy our constraints, we must build something new, or at least extend an existing system to satisfy our constraints.

We were charged with using the latest advances in human factors, end-to-end cryptography and statistical auditing techniques, while keeping costs down and satisfying many challenging constraints. We want to generate quick, verifiable tallies when the election is over, yet incorporate a variety of audit mechanisms (some voter-verifiable, others facilitated by auditors with additional privileges).

[1] A laser printer might consume 1000 watts or more while printing. A reasonably good UPS, weighing 26 kg, can provide that much power for only ten minutes. Since a printer must warm up for each page when printed one-off (perhaps 10 seconds per page), the battery might be exhausted by printing as few as 60 ballots.

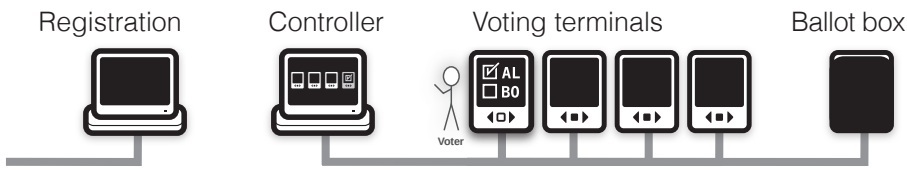

Figure 14.1: The design of the STAR-Vote system. The voter registration system (left) is connected to the Internet but not to the internal LAN. Voters move left to right. First, the voter's registration is validated, and a thermal printout indicates the proper ballot style. This moves to the controller, which scans it and issues the voter a PIN, again printed on thermal paper. The voter proceeds to any open voting terminal, enters the PIN, and is given the proper ballot style. The ballot summary is printed, and deposited into the ballot box (right).

We notably have chosen to design STAR-Vote without explicitly worrying about the constraints of state or federal certification. Of course, for STAR-Vote to go into production, these challenges need to be addressed, but at least for now, our focus has been on designing the best possible voting system given our constraints.

14.2 Voter Flow

Figure 14.1 shows how STAR-Vote works from the perspective of a voter. The STAR-Vote voting system bears a resemblance to the Hart InterCivic eSlate system and to VoteBox [510], in that the voting machines are networked together, simplifying the movement of data. Like eSlate, our design contains a networked group of voting machines that share a common judge's station with a computer like Hart InterCivic's "Judge Booth Controller" (JBC) that manages everything.

1. *Registration (poll book).* The first step for the voter is to check in with a poll worker. This is where voter registration is verified and the voter's precinct and ballot style are determined. The registration system, via cellular modem, notifies a centralized database of the voter's change in status, to eliminate any risk of double-voting.

   The registration system will use a thermal label printer to generate a sticker with the voter's name, precinct and ballot style indicated. The precinct and ballot style are also indicated with a 1-D barcode. This sticker goes into a poll book which the voter signs, providing a backup to the online database. The barcode can also be read by an off-the-shelf scanner connected to the controller. This represents the only data flow from the outside world into the internal voting network, and helps avoid data entry errors that might come from human transcription. Nothing in the barcode is secret nor is it unique to the voter. Consequently, the flow of this information does not compromise the

voter's privacy, so long as the voter is not the only voter with the same precinct and ballot style to vote at that polling location.

Provisional voters will be indicated with a suitable prefix to their precinct code, allowing the voting system to suitably distinguish their ballots from regular ones. (Provisional votes are cast by voters who, for whatever reason, do not appear in the voter registration database, and believe this to be in error. They are only tabulated after the voter's registration status is verified, typically not until at least a few days after the end of voting.)

2. *Controller.* The controller scans the barcode on the sticker to identify the voter's precinct and ballot style. The controller then prints a 5-digit code, unique for the remainder of the election in this polling place. Holding this printout, the voter can then approach any open voting terminal, enter the code, and be presented with the correct ballot style. (There will probably need to be a special alternative for ADA compliance as not all voters can see or handle paper. One possible solution is that a poll worker could enter the relevant code, then depart before the voter begins voting.)

   There are only ever a small number of 5-digit codes active at any one time, reducing the odds of a voter successfully guessing an active code and casting multiple ballots. We note that there will be no record binding the 5-digit code to the voter, helping ensure voter anonymity. We also note that these codes reduce the attack surface, relative to other voting systems that use smartcards or other active electronic devices to initialize a voting machine for each voter.

3. *Voting terminals.* The voter makes selections with the GUI (for sighted voters) or auditory UI (for non-sighted voters). There is a review screen (or the auditory equivalent) so that the voter can confirm all selections before producing a paper record.

4. *Print.* When the voter has finished making selections, the voting terminal prints two (possibly joined) items: (1) a paper ballot which includes a human-readable summary of the voter's selections and a random (non-sequential) serial number, and (2) a take-home receipt that identifies the voting terminal used, the time of the vote, and a short (16-20 character) hash code that serves as a commitment to the vote but does not reveal its contents.[2] The voting terminal also sends data about the vote and receipt to the judge's station. (See Section 14.6 for the exact cryptographic design.)

5. *Review printed record.* The voter may then review the printed record to confirm the indicated selections. There will be at least one offline station available that can scan the paper record and read it back to the voter for those who cannot visually read the paper record.

[2] A secondary hash code with as many as 16-20 additional characters may be included for additional assurance.

6. *Option: Cast or challenge/spoil.* After reviewing the ballot, the voter has a choice: Cast the ballot or spoil it. A voter might spoil the ballot because of an error (or change of heart) or because the voter wishes to challenge the voting terminal, demanding it to show that the voter's selections were correctly recorded and committed to. This process represents a novel variant on Benaloh challenges [84, 85]; rather than asking the voter a "cast or challenge" question, the voter either deposits the ballot in the box or not. This represents a potentially significant usability gain over prior variants of the Benaloh challenge.

   The two procedures are described below. Note also that there is a special procedure for provisional ballots.

   Regardless, the voter may keep the take-home paper receipt. We note that most thermal printers include a cutting device that leaves a small paper connection between the two sides of the cut. It is therefore a simple matter for the voting terminal to print a single sheet that the voter can easily separate into the ballot summary and the take-home receipt. We also note that "privacy sleeves" (i.e., simple paper folders) can protect the privacy of these printed ballots as voters carry them from the voting machine either to the ballot box to be cast, or to the judge's station to be spoiled.

   (a) *Ballot box: cast ballot.* A voter who wishes to cast the ballot takes the paper ballot summary to the ballot box. The ballot box has a simple scanner that can read the serial number from the ballot (the serial number might also be represented as a one-dimensional barcode for reliability) and communicate this to the controller, allowing the controller to keep a record of which ballots have found their way to the ballot box, and thus, which ballots should be tabulated. *An electronic ballot record is not considered complete and should not be included in the tally unless and until its corresponding paper ballot summary has been deposited in the ballot box.*

   (b) *Spoil ballot.* If the paper record is to be spoiled, the voter returns to a poll worker. The ballot serial number is scanned so that the controller can record that the ballot is to be spoiled. This informs the controller that the corresponding encrypted ballot record should not be included in contest results. Instead, it should be decrypted and published as such after the election is over. The original printed paper ballot thus corresponds to a *commitment* by the voting machine, before it ever "knew" it might be challenged. If the voting machine cannot produce a suitable proof that the ballot encryption matches the plaintext, then it has been caught cheating. Voters who don't care about verification can simply restart the process. For voters who may feel uncomfortable with this process, as it might reveal their intent to a poll worker, we note that voters could deliberately spoil ballots that misstate their true intent. We note that dedicated election monitors could be allowed to use voting machines, producing printed ballots that they would be forbidden from placing in the ballot box, but

which would be spoiled and then the corresponding ciphertext would be decrypted. In effect, election monitors can conduct *parallel testing in the field* on any voting machine at any time during the live election.

(c) *Provisional ballot.* In the case of a provisional ballot, the voter must return the printed ballot to a poll worker. The voter can choose to spoil the ballot and re-vote or to cast the ballot provisionally by having it placed—under an identifying seal—into a distinct provisional ballot box. The voter may retain the receipt to see if the ballot ends up being counted. Because the ballot box is connected to the controller over the LAN, it can also query the controller as to whether the ballot is provisional. In the event that a voter accidentally puts a provisional ballot into the ballot box, the scanner can detect this and reject the printed ballot. (Provisional ballots need to go into dedicated envelopes that are processed after the voting has ended.)

7. *At home (optional): Voter checks crypto.* The encrypted votes will be posted on a public "bulletin board" (i.e., a website maintained by the county). The voter receipt corresponds to a cryptographic hash of the encrypted vote. A voter should be able to easily verify that this vote is present on the bulletin board. If a voter spoiled a ballot, that should also be visible on the bulletin board together with its decrypted selections. This allows independent observers to know which ballots to include in the tally and allows independent verifiers to check that all spoiled ballots are correctly decrypted. Individual voters can check, without any mathematics, that the decryptions of their own spoiled ballots match their expectations.

While this process is more cumbersome than a traditional DRE voting system, it has several advantages. By having the paper elements, this system not only benefits from sophisticated end-to-end cryptographic techniques (described in Section 14.6), it also can be audited post-election, by hand, using a risk-limiting audit (see Section 14.5). Voters also have the confidence that comes from holding, verifying and casting a tangible record of their votes, whether or not they trust the computers.

14.3 Design

From the perspective of voters, the process of registration and poll-station sign-in is unchanged from current practice. Once authorized, voters proceed to a voting terminal where they use a rich interface that prevents overvotes, warns of undervotes and supports alternative input/output media for disabled and impaired voters. The printed ballot summary and the corresponding electronic ballot record both include a variety of cryptographic features, which we now describe.

14.3.1 Crypto Overview

From the perspective of election officials, the first new element in the election regimen is to generate the cryptographic keys. A set of election trustees is designated as key holders and a threshold number is fixed. The functional effect is that if there are n election trustees and the threshold value is k, then any k of the n trustees can complete the election, even if the remaining $n-k$ are unavailable. This threshold mechanism provides robustness while preventing any fewer than k of the trustees from performing election functions that might compromise voter privacy. Threshold cryptosystems are straightforward extensions of traditional public-key cryptosystems [199].

The trustees each generate a key pair consisting of a private key and a public key; they publish their public keys. A standard public procedure is then used to compute a single public key from the n trustee public keys such that decryptions can be performed by any k of the trustees. This single election public key K is published and provided to all voting terminals together with all necessary ballot style information to be used in the election. A start value z_0, which is unpredictable and unique to the election, is also chosen and distributed to each voting terminal for reasons discussed below.

During the election, voters use voting terminals to make their selections. Once selections are completed, the voting terminal produces paper printouts of two items. The first is the paper ballot summary which consists of the selections made by the voter and also includes a random (non-sequential) serial number. The second is a receipt that consists of an identification number for the voting terminal, the date and time of the vote and a short hash of the encryption of the voter's selections together with the previous hash value. Specifically, if the voter's selections are denoted by v, the i^{th} hash value produced by a particular voting terminal m in an election is computed as

$$z_i = H(E_K(v), m, z_{i-1})$$

where H denotes the hash function and E denotes encryption. This separation of the ballots into two parts makes sure that the ballot summary does not contain any voter-related information, while the take-home receipt does not leak any information about the voter choices. Furthermore, since we only store votes in an encrypted form, and since the decryption keys are kept out of the system, there is no problem with storing the votes with timestamps: they could only allow linking a voter to a ciphertext that will never be decrypted, which is harmless.

The voting terminal should retain both the encrypted ballots and the current hash value. At the conclusion of the election (if not sooner), the encrypted ballots should be posted on a publicly-accessible web page and digitally signed by the election office using a simple signature key (not the key generated by the trustees). The posting of each encrypted ballot should also include a non-interactive zero-knowledge (NIZK) proof that the ballot is well-formed. Once they receive their ballot summaries and take-home receipts, voters may either deposit their ballot summaries into a ballot box or take them to a poll worker and have them spoiled. Ballot summaries deposited in a ballot box have their serial numbers scanned and recorded. The electronically

stored encrypted vote is not considered complete (and not included in the tally) unless and until its corresponding serial number has been recorded in the ballot box.

Any electronically stored encrypted ballots for which no corresponding serial number has been scanned and recorded are deemed spoiled. The published election record should include all spoiled ballots as well as all cast ballots, but for each spoiled ballot the published record should also include a verifiable decryption of the ballot's contents. Voters should be able to easily look up digitally-signed records for any receipts they hold and verify their presence and, for spoiled receipts, the ballot contents.

A voter who takes a completed paper ballot summary to a poll worker can request that the poll worker spoil the ballot and give the voter an opportunity to re-vote. The poll worker marks both the take-home receipt and the paper ballot summary as spoiled (including removing or marking the serial number so that it will not be recorded if subsequently placed in the ballot box) and returns the spoiled ballot summary to the voter.

Upon completion of the election, the election office homomorphically combines the cast ballots into an aggregate encryption of the election tally (this can be as simple as a multiplication of the public encrypted ballots). At least k of the election trustees then each perform their share of the decryption of the aggregate as well as individual decryptions of each of the spoiled ballots. The trustees also post data necessary to allow observers to verify the accuracy of the decryptions.

A privacy-preserving risk-limiting audit is then performed by randomly selecting paper ballot summaries and matching each selected ballot with a corresponding encrypted ballot to demonstrate the correct matching and provide software-independent evidence of the outcome [488, 367, 541].

14.3.2 Triple Assurance

Three lines of evidence are produced to support each election outcome [541]. The homomorphic tallying process proves that the announced tally corresponds to the posted encrypted ballot records. The ballot challenge and receipt checking processes allow voters to check that these encrypted ballot records correctly reflect their selections. The risk-limiting audit process serves to verify the correspondence between the paper records and the electronic records. In addition, the paper records remain available in case of systemic failure of the electronic records or if a manual count is ever desired. The paper and electronic records are conveyed to the local election office separately, providing additional physical security of the redundant audit trail.

The design of the election system ensures that all three of these checks should be perfectly consistent. There is sufficient information in the records so that if any discrepancies arise (for instance because of loss of some of the electronic or paper records), the discrepancies can be isolated to individual ballots that are mismatched or counted differently.

Why combine E2E with risk-limiting auditing? Each provides different guarantees and they support each other's strengths. E2E techniques, for example, provide cryptographically strong evidence that a voter's receipt corresponds to a ballot, on the bulletin board, which has been included correctly in the final tally—a guarantee that risk-limiting audits alone cannot accomplish. However, if there's a discrepancy, E2E techniques cannot necessarily identify where things went wrong. Risk-limiting audits provide a backstop to prevent cryptographic failures from ruining the election outcome. They also provide a secondary check against machines that might be producing paper and electronic records that disagree, even if voters aren't bothering to conduct E2E challenge audits.

14.3.3 Software and Hardware Engineering

An important criteria for STAR-Vote is that it should leverage commodity components whenever feasible. This reduces cost and simplifies the ability for an election administrator to replace aging hardware by sourcing it from multiple vendors. While this paper isn't intended to cover certification issues, the separation of hardware and software allows for the possibility of *commercial off-the-shelf* (COTS) hardware, which *could* be subject to a lower bar for certification than the software.

Ideally, the voting terminals and the judge station could use identical hardware. In particular, we believe that a reasonable target might be "point of sale" terminals. These are used in restaurants worldwide. They are used in relatively demanding environments and, on the inside, are ordinary PCs, sometimes built from low-power laptop-class parts. The only missing hardware from a COTS point of sale terminal, relative to our needs for STAR-Vote, are a printer and a battery.

If you want a reliable, low-power printer, without having to worry about consumable ink or toner, there's only one choice: thermal printers. They come in a variety of widths, up to US Letter size. Thermal paper, particularly higher cost thermal paper, can last for years in an air-conditioned warehouse, although some experimentation would be required to see whether it can survive an un-air-conditioned trip in a hot car in the summer. Every shipping label from major online vendors like Amazon is printed thermally, lending some credence to its survivability in tough conditions.

Specifying a battery is more complicated. We could require that the voting terminal have an internal (and removable) battery, but this eliminates COTS point of sale terminals. Tablet computers come with built-in batteries that, at least in some cases, can last all day. Tablet computers have smaller screens than we might prefer, but they don't have hardware Ethernet ports or enough USB ports to support accessibility devices and printers.[3] Also, we would prefer to use wired networks, rather than the wireless networks built into most tablets. We note that a number of vendors are now releasing touchscreen-enabled laptops and larger touchscreen desktop models to

[3] While a single USB port can connect to a USB hub, which would then have more expandability, a *powered* USB hub might be necessary to drive some devices like a USB Ethernet adapter, complicating our requirement to keep STAR running even when on battery power.

support Windows 8. This new hardware is likely to provide good options for running STAR.

For the software layer, we see no need for anything other than a commodity operating system, like Linux, which can be stripped of unessential features to reduce the attack surface. For example, we don't need a full-blown window system or 3D graphics pipeline. All we need are basic pre-rendered ballots, as in pVote [584, 583] or VoteBox [510]. We would specify that the voting system software be engineered in a type-safe language like Java or C# (eliminating buffer overflow vulnerabilities, among other problems) and we would also specify that the software be engineered with *privilege separation* [473], running separate parts of the voting software as distinct applications, with distinct Unix user-ids, and with suitably reduced privileges. For example, the storage subsystem can maintain append-only storage for ballots. The voter-facing UI would then have no direct access to ballot storage, or the network, and could be "rebooted" for every voter. Consequently, a software compromise that impacts the UI application could impact at most one voter. A tablet that includes a Trusted Platform Module (TPM) can provide additional assurance that the correct software — and only the correct software — is running on the device.

A separation architecture like this also provides some degree of protection over sensitive cryptographic key materials, e.g., if we want every voting terminal to have a unique private key to compute digital signatures over ballots, then we must restrict the ability for compromised software to extract the private keys. DStar [587], for example, used this technique to protect the key material in an SSL/TLS web server.

14.4 Usability

14.4.1 Design Considerations

In designing this reference voting system it was important to maximize the usability of the system within the framework of enhanced security and administrative expediency. The overall design of the system was strongly influenced by usability concerns. For example, a proposal was put forth to have all voters electronically review the paper record on a second station; this was rejected on usability grounds. ISO 9241 Part 11 [319] specifies the three metrics of usability as effectiveness, efficiency and satisfaction, and these are the parameters we attempt to maximize in this design. Effectiveness of the system means that users should be able to reliably accomplish their task, as they see it. In voting, this means completing a ballot that correctly records the candidate selections of their choice, whether that be through individual candidate selection by race, straight party voting, or candidate write-ins. Efficiency measures the ability of a voter to complete the task with a minimum of effort, as measured through time on task or number of discrete operations required to complete a task. Efficiency is important because users want to complete the voting task quickly and voting officials are concerned about voter throughput. Reduced efficiency means longer lines

for waiting voters, more time in the polling booth, and higher equipment costs for election officials. Satisfaction describes a user's subjective assessment of the overall experience. While satisfaction does not directly impact a voter's ability to cast a vote in the current election, it can have direct impact on their willingness to engage in the process of voting at all, so low satisfaction might disenfranchise voters even if they can cast their ballots effectively and efficiently. How does this design seek to maximize these usability metrics? For voting systems, the system must generally be assumed to be walk-up-and-use. Voting is an infrequent activity for most, so the system must be intuitive enough that little to no instruction is required to use it. The system should minimize the cognitive load on voters, so that they can focus on making candidate selections and not on system navigation or operation. The system should also mitigate the kinds of error that humans are known to make, and support the easy identification and simple correction of those errors before the ballot is cast.

Why Not Paper?

Paper ballots (bubble ballots in particular) have many characteristics that make them highly usable [228, 127]. Users are familiar with paper, and most have had some experience with bubble-type item selection schemes. Voting for write-in candidates can be relatively simple and intuitive. Unlike electric voting machines, paper is nearly 100% reliable and is immune from issues of power interruption. Further, paper leaves an auditable trail, and wholesale tampering is extremely difficult. However, paper is not a perfect solution. Voters actually show higher satisfaction with electronic voting methods than they do with paper [226] and paper has significant weaknesses that computers can overcome more easily. First, the ambiguity that can be caused by partial marks leads to substantial problems in counting, recounting and re-interpreting paper ballots. Second, voting by individuals with disabilities can be more easily accommodated using electronic voting methods (e.g., screen readers, jelly switches). Third, electronic voting can significantly aid in the reduction of error (e.g., undervotes, overvotes, stray marks) by the user in the voting process. Fourth, electronic voting can more easily support users whose first language is not English, since additional ballots for every possible language request do not have to be printed, distributed and maintained at every polling location. This advantage is also evident in early voting and vote center administration; rather than having to print, transport, secure and administer every possible ballot for every precinct, the correct ballot can simply be displayed for each voter. Computers also facilitate sophisticated security and cryptography measures that are more difficult to implement in a pure paper format. Finally, administration of the ballots can be easier with electronic formats, since vote counting and transportation of the results are more efficient. We have taken a hybrid approach in this design, by using both paper and electronic voting methods in order to create a voting system that retains the benefits of each medium while minimizing their weaknesses.

Usability vs. Security

Usability and security are often at odds with each other. Password design is a perfect example of this tension. A system that requires a user have a 32-character password with upper and lower case letters, digits and symbols with no identifiable words embedded might be highly secure, but it would have significant usability issues. Further, security might actually be *compromised* since users are likely to write such a difficult password down and leave it in an insecure location (e.g., stuck to the computer monitor). For voting systems, we must strive for maximum usability while not sacrificing the security of the system (our security colleagues might argue that we need to maximize security while not sacrificing usability). In our implementation, many of the security mechanisms are invisible to the user. Those that are not invisible are designed in such a way that only those users who choose to exercise the enhanced security/verifiability of the voting process are required to navigate additional tasks (e.g., ballot challenge, post-voting verification).

Accessibility vs. Security

STAR-Vote makes strategic use of paper to enhance the overall security and auditability of the voting process. From an auditability standpoint, the presence of the paper ballot allows matching of the paper and electronic records and preserves a separate physical copy apart from the electronic tally. From a security standpoint, it allows a voter to verify that the choices selected on the electronic voting terminal (DRE) have been faithfully recorded on the paper ballot (although this voter verification is not as robust as one might hope [227]), and challenge their vote if they choose to do so. However, the added benefits provided by the inclusion of paper come at a cost to the accessibility of the system. Visually impaired voters must now be given a way to verify the contents of printed material and be guided in the handling of that paper into the scanners and ballot boxes. Voters with mobility impairments must now handle these paper ballots with moderate dexterity in order to feed them into the scanning ballot boxes as well. Solutions to this trade-off are still under evaluation. Many obvious solutions, such as giving voters with disabilities the option to simply cast an electronic ballot without a paper record, seriously compromise the overall security and auditability of the voting system, and also present significant privacy concerns, since voters who opt out of the main flow might be easily identified. Simple but non-optimal solutions are being considered (test-to-speech scanning stations, ballot privacy sleeves and increased poll worker involvement), but we continue to investigate more elegant solutions that involve automatic paper handling mechanisms. A final design has still not been identified.

Error Reduction

The use of computers in combination with paper is anticipated to reduce errors committed by voters. Because voters will fill out the ballot on electronic voting terminals, certain classes of errors are completely eliminated. For example, it will be impossible

to over vote or make stray ballot marks, as the interface will preclude the selection of more than a single candidate per race. Under voting will be minimized by employing sequential race presentation, forcing the voter to make a conscious choice to skip a race [268]. Under votes will also be highlighted in color on the review screen, providing further opportunity for users to correct accidental under votes. This review screen will also employ a novel party identification marker (see below) that will allow a voter to easily discern the party for which they cast a vote in each race. The use of the paper ballot (printed when the voter signals completion) provides the voter with a final chance to review all choices before casting the final ballot.

14.4.2 User Interface Design Specification

The basic design for the UI is a standard touchscreen DRE with auditory interface for visually impaired voters and support for voter-supplied hardware controls for physical impairments (e.g., jelly switches).

The VVSG

The starting point for UI specifications is the 2012 draft version 1.1 of the Voluntary Voting System Guidelines (VVSG). These guidelines specify many of the critical properties required for a high-quality voting system user interface, from simple visual properties such as font size and display contrast to more subtle properties such as ballot layout. They also require that interfaces meet certain usability benchmarks in terms of error rates and ballot completion time. We believe that no extant commercial voting UI meets these requirements, and that any new system that could meet them would be a marked improvement in terms of usability. That said, there are some additional requirements that we believe should be met.

Accessibility

While the VVSG includes many guidelines regarding accessibility, more recent research aimed at meeting the needs of visually-impaired voters [462] has produced some additional recommendations that should be followed. These include:

- In order to capitalize on user preference, a synthesized male voice should be used.
- Navigation should allow users to skip through sections of speech that are not important to them as well as allowing them to replay any parts they may have missed or not comprehended the first time.
- At the end of the voting process, a review of the ballot must be included, but should not be required for the voter.

Review Screens

Another area where the VVSG can be augmented concerns review screens. Voter detection of errors (or possible malfeasance) on review screens is poor [227], but there is some evidence that UI manipulations can improve detection in some cases [132]. Thus, STAR-Vote requires the following in addition to the requirements listed in the VVSG:

- Full names of contests and candidates should be displayed on the review screen; that is, names should be text-wrapped rather than truncated. Party affiliation should also be displayed.
- Undervotes should be highlighted using an orange-colored background.
- Activating (that is, touching on the visual screen or selecting the relevant option in the auditory interface) should return the voter to the full UI for the selected contest.
- In addition to party affiliation in text form, graphic markings should be used to indicate the state of each race: voted Republican, voted Democratic, voted Green, etc.—with a distinctive graphic for "not voted" as well. These graphic markings should be highly distinguishable from each other so that a rapid visual scan quickly reveals the state of each race, while taking note of potential usability issues with graphics symbols [533]. Exact graphic symbols for STAR-Vote have not yet been determined.

Paper Record

The VVSG has few recommendations for the paper record. For usability, the paper record should meet VVSG guidelines for font size and should contain full names for office and candidate. To facilitate scanner-based retabulations, the font should be OCR-friendly. Contest names should be left-justified while candidate names should be right-justified to a margin that allows for printing of the same graphic symbols used in the review screen to facilitate rapid scanning of ballots for anomalies. Candidate names should not be placed on the same line of text as the contest name and a thin horizontal dividing line should appear between each office and the next in order to minimize possible visual confusion.

14.4.3 Issues That Still Need to Be Addressed

There are still several issues that need to be addressed in order to make the system have the highest usability. The first of these is straight party voting (SPV). SPV can be quite difficult for a voter to understand and accomplish without error, particularly if voters intend to cross-vote in one or more races [131]. Both paper and electronic methods suffer from these difficulties, and the optimum method of implementation will require additional research. Races in which voters are required to select more

than one candidate (k of n races) also create some unique user difficulties, and solutions to those problems are not yet well understood.

14.5 Audit

The E2E feature of STAR-Vote enables individual voters to confirm that their votes were included in the tabulation, and that the encrypted votes were added correctly. The challenge feature, if used by enough voters, assures that the encryption was honest and that substantially all the votes are included in the tabulation. But there might not be many voters who challenge the system; the voters who do are hardly representative of the voting public; and some problems may go unnoticed. Moreover, the anonymized form of E2E used here does not allow a voter to confirm that *others'* ballots were included in the tabulation, only that those ballots that were included were included correctly.

The paper audit trail enables an entirely independent check that the votes were included and tabulated accurately, that the visible trace of voter intent as reflected in the ballot agrees with the encryption, and, importantly, that the winners reported by the voting system are the winners that a full hand count of the audit trail would reveal. The key is to perform a compliance audit to ensure that the audit trail of paper ballots is adequately intact to determine the outcomes, and then to perform a risk-limiting audit of the machine interpretation against a manual interpretation of the paper ballots. For the risk-limiting audit, STAR-Vote uses SOBA [87] with improvements given by [367].

A risk-limiting audit guarantees a large minimum chance of a full hand count of the audit trail if the reported outcome (i.e., the set of winners) disagrees with the outcome that the full hand count would reveal. The full hand count then sets the record straight, correcting the outcome before it becomes official. Risk-limiting audits are widely considered best practice for election audits [366, 114].

The most efficient risk-limiting audits, ballot-level comparison audits, rely on comparing the machine interpretation of individual ballots (cast vote records or CVRs) against a hand interpretation of the same ballots [540, 87, 367]. Current federally certified voting systems do not report cast vote records, so they cannot be audited using the most efficient techniques [367, 541]. This necessitates expensive work-arounds.[4] The preamble to conducting a ballot-level comparison audit using currently deployed voting systems can annihilate the efficiency advantage of ballot-level comparison audits [541].

A big advantage of STAR-Vote is that it records and stores individual cast vote records in a way that *can* be linked to the paper ballot each purports to represent,

[4]For instance, a *transitive audit* might require marking the ballots with unique identifiers or keeping them in a prescribed order, re-scanning all the ballots to make digital images, and processing those images with software that can construct CVRs from the images and associate the CVRs with the ballots. That software in turn needs to be programmed with all the ballot definitions in the contest, which itself entails a great deal of error-prone handwork.

through encrypted identifiers of the ballot corresponding to each voter's selections, separately for each contest. This makes ballot-level comparison audits extremely simple and efficient. It also reduces the vulnerability of the audit to human error, such as accidental changes to the order of the physical ballots.[5]

A comparison audit can be thought of as consisting of two parts: Checking the addition of the data,[6] and randomly spot-checking the accuracy of the data added, to confirm that they are accurate enough for their tabulation to give the correct electoral outcome. The data are the votes as reported by the voting system. For the audit to be meaningful, the election official must commit to the vote data before the spot-checking begins. Moreover, for the public to verify readily that the reported votes sum to the reported contest totals, it helps to publish the individual reported votes. However, if these votes were published ballot by ballot, pattern voting could be used to signal voter identity, opening a communication channel that might enable widespread wholesale coercion [482, 87].

The SOBA risk-limiting protocol [87] solves both of these problems: It allows the election official to commit cryptographically and publicly to the vote data; it publishes the vote data in plain text but "unbundled" into separate contests so that pattern voting cannot be used to signal. Moreover, the computations that SOBA requires are extremely simple (they are simplified even further by [367]). The simplicity increases transparency, because observers can confirm that the calculations were done correctly with a pencil and paper or a hand calculator.

The encrypted ballot/contest identifiers on the ballot that STAR-Vote produces allow the electronic cast vote records for each contest to be linked to the paper they purport to represent. This simplifies SOBA procedures because it eliminates the need to store ballots in a rigid order. Moreover, because the voting terminal generates both the electronic vote data and the paper ballot, the audit should find very few if any discrepancies between them.

But since voters and election workers will handle the ballots in transit from the voting terminal to the scanner to the audit, voters might make marks on their ballots. Depending on the rules in place for ascertaining voter intent from the ballot, those marks might be interpreted as expressing voter intent different from the machine-printed selections, in which case the SOBA audit might find discrepancies.

It could also happen that a ballot enters the ballot box but its serial number is not picked up, so the electronic vote data ends up in the "untallied but unspoiled" group. This should be detectable by a compliance audit [87, 367, 541] as a mismatch between the number of recorded votes and the number of pieces of paper, providing an opportunity to resolve the problem before the audit begins.

[5] For instance, we have seen groups of ballots dropped on the floor accidentally; even though none was lost, restoring them to their original order was impossible.

[6] This presupposes that the contest under audit is a plurality, majority, super-majority, or vote-for-k contest. The operation that must be checked to audit an instant-runoff contest is not addition, but the same principle applies.

If such cases remain and turn up in the audit sample, SOBA would count them as discrepancies and the sample might need to expand, either until there is strong evidence that the electoral outcomes are correct despite any errors the audit uncovers, or until there has been a complete hand count.

The random selection of ballots for the SOBA audit should involve public participation in generating many bits of entropy to seed a high-quality, public, pseudo-random number generator (PRNG), which is then used to select a sequence of ballots to inspect manually [367]. (For instance, audit observers might roll 10-sided dice repeatedly to generate a 20-digit number.) Publishing the PRNG algorithm adds transparency by allowing observers to verify that the selection of ballots was fair.

14.6 The Cryptographic Workflow

The Core Elements

At its core, the cryptographic workflow of STAR-Vote follows the approach of Cramer, Gennaro and Schoenmakers [178], also used in Helios [46] and Vote-Box[510], among others. Cryptographic analyses of this approach can be found in [99, 171]. We then augment this approach in various ways in order to ease the detection of and recovery from potential problems.

STAR-Vote keeps an electronic record of all the votes encrypted with a threshold cryptosystem (so that decryption capabilities are distributed to protect voter privacy) that has an additive homomorphic property (to allow individual encrypted ballots to be combined into an aggregate encryption of the tally). The common exponential version of the ElGamal cryptosystem [213] satisfies the required properties, and stronger security is obtained by using PPATS encryption [186], in particular against key manipulation errors by the trustees and long-term security. The encryption scheme comes with an extraction function Ext that, from a ciphertext, extracts a commitment on the encrypted value. In the case of ElGamal, this commitment is the ciphertext itself, while in the case of PPATS, it is a perfectly hiding homomorphic commitment.

Cryptographic key generation can be accomplished in one of two ways, depending on the availability of the election trustees and the desired amount of robustness. The preferred process offers general threshold key generation requiring multiple rounds (see [252] for ElGamal and PPATS), but can be simplified into a single-round solution if redundancy is eliminated (as in Helios for instance [46]). At the end of the key generation procedure, the trustees each hold a private key share that does not contain any information on the full private key, and the unique public key K corresponding to those shares is published.

During the polling phase, the ballot-marking devices encrypt the votes of each voter using the public key K. This encryption procedure is randomized in order to

make sure that two votes for the same candidates result in ciphertexts that look independent to any observer.

Following Benaloh [84], a cryptographic hash value of the commitment extracted from each ciphertext (and of a few more data, as discussed below) is also computed, fingerprinting the ballot to a 256-bit string. An abridged form of this is provided to the voter in a human readable form as part of the take-home receipt. All the hashes and commitments are computed and posted on a publicly accessible web page, as soon as the polls are closed. This web page is digitally signed by the election office using a traditional signature key (as performed by [46]). This signature operation makes it infeasible to consistently modify the content of the web page without the help of the signer, and provides evidence against a malicious signer who would try to sign various versions of the bulletin board.

The posting of all the hashes gives all voters the ability to verify that their ballots have been recorded properly. The commitments can also be checked for consistency with the hashes and used to confirm the homomorphic aggregation of the individual ballots into a single encryption of the sum of the ballots, which constitutes an encryption of the election tallies.

At the end of the election, any set of trustees that achieve the pre-set quorum threshold use their respective private keys to decrypt the derived aggregate tally encryption. This procedure is simple and efficient and can be completed locally without interaction between the trustees. We note that the individual encrypted ballots, from which the aggregate encryption of the tallies is formed, are never individually decrypted. However, each spoiled ballot *is* individually decrypted using exactly the same process that is used to decrypt the aggregate tally encryption.

The elements we just described make the core of the workflow and are sufficient to compute an election tally while preserving the privacy of the votes. We now explain various ways in which this simple workflow is hardened in order to make sure that the tally is also correct. All the techniques that follow enable the verification of different aspects of the ballot preparation and casting.

Hardening Encryption

Since the tally does not involve the decryption of any individual ballot, and since the audit procedure relies on the fact that all counted ballots are properly formed, it is crucial to make sure that all the encrypted ballots that are added correspond to valid votes [161]. This is achieved by requiring the ballot-marking devices to compute, together with the encryption of the votes, a non-interactive zero-knowledge (NIZK) proof that each ballot is well-formed. Such a proof guarantees that each ciphertext encrypts a valid vote and does not leak any other information about the content of the vote. As a side benefit, this proof can be designed to make the ballots non-malleable, which provides an easy technique to prevent the replay of old ballots (i.e., reject duplicates). Traditional sigma proofs provide the required security properties and are described and analyzed in [99].

We note that, if malicious software were to get into the voting system, it could use the randomness inherent in the encryption process to encode a subliminal message to an external observer. This sort of threat, along with the threat of a malicious voting machine that simply records every vote cast, in plaintext, in internal memory, is something that cryptography cannot address. (More discussion on this appears in Section 14.3.3.)

Hardening Decryption

Making sure that the encrypted ballots are valid is not enough: we also need to make sure that the tally is correctly decrypted as a function of those encrypted ballots: otherwise, malicious trustees (or trustees using corrupted devices) could publish an outcome that does not correspond to these ballots. As a result, we require the trustees to provide evidence of the correctness of the decryption operations that they perform. This can also be accomplished with sigma proofs in the case of ElGamal or more simply by publishing commitment openings in the case of PPATS.

Hardening the Timeline

The procedures described above prevent malfunctioning or corrupted voting terminals or trustees to falsify individual ballots or decryption operations.

The detection of manipulation of encrypted ballots can be more effective by linking ballots with each other, using hash chaining [511, 88]. For this purpose, each ballot marking device is seeded, at the beginning of the election, with a public start value z_0 that includes a unique identifier for the election. This unique identifier is chosen at random shortly before the election, either in a central way or by the poll workers themselves at the beginning of election day.

From this seed, all election events are chain hashed, with z_{i+1} being computed as a hash of z_i concatenated to the id of the machine on which the event happens and to the event content. Two such chains are maintained and properly separated. One is internal and contains the full election data, including the encryption of the votes, the casting time of each paper ballot, and information on machines being added or removed. The second is public and chains the commitment extracted from all encrypted votes, together with time and identifiers for the election and voting machine. The public hash is the one actually printed on the take-home receipt. When the polls close, the final value of the hash chains are digitally signed, and the public chain is made public together with all the information needed for its reconstruction.

As a result of this procedure, any removed ballot will invalidate the hash chain which is committed to at the close of the election and whose constituents appear on the voter take-home receipts.

Hardening the Link between the Paper and Electronic Election Outcome

As described in Section 14.5, STAR-Vote includes a risk-limiting audit (RLA) based on the human-readable versions of each ballot summary printed by the voting terminals and inspected for correctness by voters. This RLA comes in addition to the cast or challenge procedure discussed above, and the production of the inputs for the RLA is an original contribution of STAR-Vote.

The requirement for running the RLA is to commit on a full electronic record including a 1-to-1 mapping and evidence that this electronic record leads to the announced outcome. This is achieved as follows.

1. For each ballot, the ballot marking device selects a random ballot id sequence number *bid*. This *bid* is printed on the ballots as a barcode. Furthermore, for each race r in which the voter participates, an encryption of $H(bid|0r)$ is also computed and appended to the encryption of the choices.

2. At the end of the day, and before decryption of the tallies, the trustees (or their delegates) shuffle and rerandomize all encrypted votes, race by race. This shuffle does not need to be verifiable, even though a verifiable shuffle would improve accountability by making it possible to verify that the shufflers did not cheat if it happens that a discrepancy is detected during the RLA. However, in the case of a non-verifiable shuffle, the shufflers must save their permutation and randomness until the end of the election audit. The non-verifiable solution is preferred for its simplicity (verifiable shuffles are particularly challenging to implement properly) and for its efficiency (permutations and reencryption factors can be precomputed, leaving only one multiplication to perform per ciphertext in the online phase, which is convenient when millions of ciphertexts have to be shuffled).

3. When the trustees decrypt the homomorphically added votes, they also decrypt the output of this shuffle. For each race, this provides a list of elements of the form $H(bid|0r)$ and the corresponding cleartext choices.

4. Now, auditors can sample the paper ballots, read the *bid* printed on them, recompute the value of $H(bid|0r)$ for all races present on the paper ballot, and compare to the electronic record (as well as check many other things, as prescribed for the risk-limiting audit).

The use of hashed *bid*'s has the important benefit of making sure that someone who does not know a *bid* value cannot, by looking at the electronic record, link the selections made for the different races on a single ballot, which protects from pattern voting attacks. There is no need for such a protection from someone who can access the paper ballots, since that person can already link all races just by looking at the paper.

The Full Cryptographic Protocol

The resulting cryptographic workflow is as follows.

1. The trustees jointly generate a threshold public key/private key encryption pair. The encryption key K is published.
2. Each voting terminal is initialized with the ballot and election parameters, the public key K and seeds z_0^p and z_0^i that are computed by hashing all election parameters and a public random salt z_0.
3. When a voter completes the ballot marking process selection to produce a ballot v, the voting terminal performs the following operations:
   (a) It selects a unique and unpredictable ballot identifier *bid*, as well as a unique (but possibly predictable) ballot casting identifier *bcid*.
   (b) It computes an encryption $c_v = E_K(v)$ of the vote, as well as a NIZK proof p_v that c_v is an encryption of a valid ballot. This proof is written in such a way that it can be verified from $Ext(c_v)$ only.
   (c) For each race $r_1, \ldots, r_n$ in which the voter takes part, it computes an encryption $c_{bid} = E_K(bid|0r_1)|0\cdots|0E_K(bid|0r_n)$.
   (d) It computes a public hash code $z_i^p = H(bcid|0Ext(c_v)|0p_v|0m|0z_{i-1}^p)$, where m is the voting terminal unique identifier, as well as an internal hash $z_i^i = H(bcid|0c_v|0p_v|0c_{bid}|0|0m|0z_{i-1}^i)$.
   (e) It prints a paper ballot in two parts. The first contains v in a human readable format as well as c_{bid} and *bcid* in a robust machine readable format (e.g., as barcodes). The second is a voter take-home receipt that includes, the voting terminal identifier m, the date and time, and the hash code z_i^p (or a truncation thereof), all in a human-readable format.
   (f) It transmits $(bcid, c_v, p_v, c_{bid}, m, z_i^p, z_i^i)$ to the judge's station.
4. When a ballot is cast, the ballot casting id *bcid* is scanned and sent to the judge's station. The judge's station then marks the associated ballot as cast and ready to be included in the tally. This information is also broadcast and added in the two hash chains.
5. When the polls are closed, the tally is computed: the product of all cast encrypted votes is computed and verifiably decrypted, providing an election result.
6. The data needed for the risk limiting audit is computed, as described above.

All the data included in the public hash chain are eventually digitally signed and published by the local authority. Those audit data are considered to be valid if the hash chain checks, if all cryptographic proofs check, that is, if the ballot validity proofs check, if the homomorphic aggregation of the committed votes is computed and opened correctly, and if all spoiled ballots are decrypted correctly.

Write-In Votes

So far, we have not described how our cryptographic construction can support write-in voting. Support for write-in votes is required in many states. To be general-purpose, STAR-Vote adopts the vector-ballot approach [346], wherein there is a separate homomorphic counter for the write-in slot plus an encryption of the string in the write-in. If there are enough write-in votes to influence the election outcome, then the write-in slots, across the whole election, will be mixed and tallied (together with the corresponding counters).

We note that, at least for elections in our state, write-in candidates must be registered in advance. It's conceivable that we could simply allocate a separate homomorphic counter for each registered candidate and have the STAR-Vote terminal help the voter select the desired "write-in" candidate. Such an approach could have significant usability benefits but is expected to require some update of regulations.

14.7 Threats

To evaluate the design and engineering of STAR-Vote, it's helpful to have a threat model in mind. The obvious place to start would be VoteBox [510], which is closely related to STAR-Vote. The original VoteBox authors did not state a concise threat model, but considered several kinds of threats and security design goals:

Software independence STAR-Vote, like VoteBox, should be able to produce a proof of the correctness of an election that does not require any statement about the correctness of the software used in STAR-Vote. VoteBox achieved this through end-to-end cryptographic means. STAR-Vote uses similar cryptography and adds a risk-limiting audit that can verify the correspondence between STAR-Vote printed ballot records and their electronic counterparts, adding a degree of flexibility if the cryptography cannot prove an exact correspondence to determine exactly what went wrong.

Reduced trusted computing base STAR-Vote, like VoteBox or any other software artifact, would benefit from having simpler code and less of it. This makes it easier to verify and less likely to have bugs. Software independence means that STAR-Vote's software is not required for *correctness* of the election outcome, but it does help defeat attacks which could disable the system, destroy records, or otherwise cause grief to election officials running STAR-Vote. VoteBox specifies that it uses pre-rendered user interfaces [583, 584]. STAR-Vote should probably use this technique as well.

Robustness against data loss STAR-Vote, like VoteBox, specifies that vote records be stored on every voting terminal in the local polling place, using tamper-evident logging techniques. STAR-Vote adds a printed ballot record, stored in a ballot box. VoteBox went a step further by considering the real-time one-way

transfer of vote records out of the polling place, across the Internet, to a central election headquarters. While STAR-Vote could add this in the future, it's not part of the initial design.

Mega attacks In the VoteBox paper, the authors considered a variety of attacks with highly capable attackers. Such attackers might run a concurrent election on parallel equipment, in an attempt to substitute the results for genuine votes. Other attackers might mount a "booth capture" attack, wherein armed gunmen take over a polling place and cast votes as fast as possible until the police arrive. These attacks, needless to say, are well within the ability of STAR-Vote's cryptographic and risk-limiting infrastructure to detect. The best such attackers can hope to do is, in effect, mount a denial of service attack against the election. Attackers with that as their goal can arrive at much simpler approaches and STAR-Vote has relatively little it can offer beyond any other election system in this regard.

A full consideration of threats to STAR-Vote and their corresponding countermeasures or mitigations would be far too long to fit in this paper. Instead, we focus on several areas where STAR-Vote differs from other E2E voting systems in the literature.

14.7.1 Coercion

In designing STAR-Vote, we made several explicit decisions regarding how much to complicate the protocol and impede the voter experience in order to mitigate known coercion threats. Specifically, one known threat is that a voter is instructed to create a ballot in a particular way but to then execute a decision to cast or spoil the ballot according to some stimulus received after the ballot has been completed and the receipt has been generated. The stimulus could come, for example, from subtle motions by a coercer in the poll site, the vibration of a cell phone in silent mode, or some of the (unpredictable) data that is printed on the voter's receipt. Some prior protocols have required that the receipt, although committed to by the voting device, not be visible to the voter until after a cast or spoil decision has been made (perhaps by printing the receipt face down behind a glass barrier) and configuring poll sites so that voters cannot see or be seen by members of the public until after they have completed all steps. We could insist on similar measures here, but in an era where cell phones with video recording capabilities are ubiquitous and eyeglasses with embedded video cameras can easily be purchased, it seems unwise to *require* elaborate measures which mitigate some coercion threats but leave others unaddressed.

14.7.1.1 Chain Voting

A similar threat of "chain voting" is possible with this system wherein a voter early in the day is instructed to neither cast nor spoil a ballot but to instead leave the poll site

with a printed ballot completed in a specified way. This completed ballot is delivered to a coercer who will then give this ballot to the next voter with instructions to cast the ballot and return with a new printed ballot—again completed as specified. Chain voting can be mitigated by instituting timeouts which automatically spoil ballots that have not been cast within a fixed period after having been printed. We also expect to have procedures in place to prevent voters from accidentally leaving poll sites with printed ballots. We note that the timeout period need only cover the time we expect will be required for a voter to cross the room with a printed ballot and place it in the box, allowing for a relatively tight time bound, probably less than 5 minutes, although we'd need to run this in practice to understand the distribution of times that might happen in the real world.

(We note that traditional paper ballots sometimes include a perforated header section which includes a serial number. A poll worker keeps one copy of this number and verifies that the ballot a voter wishes to cast matches the expected serial number. If so, the serial number is then detached from the ballot and deposited in the box. STAR-Vote could support this, but we believe it would damage STAR-Vote's usability. The timeout mechanism seems like an adequate mitigation.)

We do, however, take measures to prevent wholesale coercion attacks such as those that may be enabled by pattern voting. For instance, The SOBA audit process is explicitly designed to prevent pattern-voting attacks; and the high assurances in the accuracy of the tally are acheived without ever publishing the full set of raw ballots.

An interesting concern is that our paper ballots have data on them to connect them to electronic ballot records from the voting terminals and judge's console. The very data that links a paper ballot to an electronic, encrypted ballot creates a potential vulnerability. Since some individual paper ballot summaries will be selected for post-election audit and made public at that time, we are careful to not include any data on the voter's take-home receipt which can be associated with the corresponding paper ballot summary.

14.7.1.2 Absentee and Provisional Ballots

There are several methods available for incorporating ballots which are not cast within the STAR-Vote system, such as absentee and provisional ballots. The simplest approach is to completely segregate votes and tallies, but this has several disadvantages, including a reduction in voter privacy and much lower assurance of the accuracy of the combined tally.

It may be possible to eliminate all "external" votes by providing electronic means for capturing provisional and remote ballots. However, for the initial design of the STAR-Vote system, we have chosen to avoid this complexity. Instead, we ask that voting officials receive external votes and enter them into the STAR-Vote system as a proxy for voters. While this still does not allow remote voters to audit their own ballots, the privacy-preserving risk-limiting audit step is still able to detect any substantive deviations between the paper records of external voters and their elec-

tronically recorded ballots. This provides more supporting evidence of the veracity of the outcome without reducing voter privacy.

14.7.2 Further Analysis

If we wished to conduct a more in-depth threat modeling exercise, one place to begin would be the threat model developed by the California Top To Bottom Review's source code audit teams (see, e.g., [318]). They considered different levels of attacker access, ranging from voters to election officials. They also considered different attacker motives (disrupt elections, steal votes, coerce voters) and different attack outcomes (detectable vs. undetectable, recoverable vs. unrecoverable, prevention vs. detection, wholesale vs. retail, and casual vs. sophisticated). A complete consideration of STAR-Vote against all these criteria would take far more space than is available in this venue. Instead, we now focus on where STAR-Vote advances the state of the art in these areas.

Most notably, STAR-Vote's combination of end-to-end cryptography with risk-limiting audits of paper ballots is a game changer, in terms of thwarting attackers who might want to disrupt elections. Unlike paperless systems, STAR-Vote has the ability to fall back to the paper records, with efficient processes to detect when inconsistencies exist that would require this. This radically improves STAR-Vote's recoverability from extreme failures.

Similarly, while STAR-Vote is "software independent," we must concern ourselves with software tampering that does not change any of the cryptographic computations, but instead causes the STAR-Vote machines to silently record everything the voter does. This threat cannot be mitigated by better cryptography or ballot auditing. The only likely solution is some sort of trusted platform management (TPM), where the hardware will refuse to run third-party code (more discussion on this appears in Section 14.3.3).

Lastly, we consider a threat that only arises in E2E systems: presentation of a fraudulent voting receipt. Consider the case where a voter may spoil her ballot and take it home to verify against the public bulletin board. A malicious voter with access to similar printers could produce a seemingly legitimate ballot for which there is no correspondence on the public bulletin board, thus "proving" that the election system lost a record. Similar defaming attacks could be made by forging the receipt that a voter can take home after casting a ballot. For STAR-Vote, we have considered a number of mitigations against these attacks, ranging from cryptographic (having the voting terminals compute a digital signature, with protected key material) to procedural (e.g., watermarking the paper or having poll workers physically sign spoiled ballots). Real STAR-Vote deployments will inevitably use one or more of these mitigations.

14.8 Conclusions and Future Work

In many ways, STAR-Vote is a straightforward evolution from existing commercial voting systems, like the Hart InterCivic eSlate, mixing in advanced cryptography, software engineering, usability and auditing techniques from the research literature in a way that will go largely unnoticed by most voters, but that has huge impact on the reliability, accuracy, fraud-resistance and transparency of elections. Of course, we can also take this opportunity to improve more pragmatic features, such as offering better support for the election administration's desired workflow. Clearly, we're long overdue for election systems engineered with all the knowledge we now have available.

STAR-Vote also opens the door to a variety of interesting future directions. For example, while STAR-Vote is intended to service any given county as an island unto itself, there's no reason why it cannot also support *remote voting*, where ballot definitions could be transmitted to a remote supervised kiosk, which securely returns the electronic and paper records. By virtue of STAR-Vote's cryptographic mechanisms, such a remote vote is really no different than a local provisional vote and can be resolved in a similar fashion, preserving the anonymity of the voter. (A variation on this idea was earlier proposed as the RemoteBox extension [512] to VoteBox [510].) This could have important ramifications for overseas and military voters with access to a suitable impromptu polling place, e.g., on a military base or in a consular office.

(We do not want to suggest that STAR-Vote would be suitable for *Internet* voting. Using computers of unknown provenance, with inevitable malware infections, and without any systematic way to prevent voter bribery or coercion, would be a foolhardy way to cast ballots. A STAR-Vote variant, running in a web browser and printing a paper ballot returned through the postal mail, might well be feasible as a replacement for current vote-by-mail practices. A full consideration of this is left for future work.)

STAR-Vote anticipates the possibility that voting machine hardware might be nothing more than commodity computers running custom software. It remains unclear whether off-the-shelf computers can be procured to satisfy all the requirements of voting systems (e.g., long-term storage without necessarily having any climate control, or having enough battery life to last for a full day of usage), but perhaps such configurations might be possible, saving money and improving the voting experience.

For additional details on STAR-Vote, Travis County has recently published a detailed "request for information" (RFI)[7] which refines the specification here with additional details on every aspect of the system. They're soliciting feedback as part of the RFI process,[8] which will hopefully then lead to a subsequent "request for proposals" (RFP, also called a "tender").

[7] http://traviscountyclerk.org/eclerk/content/images/pdf_STARVote_2015.06.03_RFI.pdf

[8] http://traviscountyclerk.org/eclerk/Content.do?code=News.StarVote

Acknowledgments

Pereira's work was supported by the Belgian French Community through the SCOOP ARC project and by the European Commission through the HOME/2010/ISEC/AG/INT-011 B-CCENTRE project. Wallach and Kortum are supported, in part, by the National Science Foundation (CNS-1409401).