# Table of Contents

# Introduction

The vote is a cornerstone of all properly functioning democracies. Most modern voting technologies include some electronic component, such as an optical scanner or punch card machine, used to assist in counting votes recorded on paper. In the past decade, however, Direct Recording Electronic (DRE) voting machines have gained popularity. In DRE systems, votes are recorded through a computer interface, and votes remain in electronic form throughout the election. In the United States, adoption of DRE machines has grown from approximately 25 percent of all voting devices in 2004 [1] to over 50 percent of devices in 2012 [2]. The same trend exists internationally: both Brazil and India have held fully electronic elections since 2002 and 2004, respectively [3].

In theory, direct recording electronic systems reduce election overhead by reducing the time required to tabulate votes and eliminating the need to print, distribute, and discard the millions of paper ballots generated by traditional optical scan or punch card elections. They also enable additional assistive technologies that would otherwise be costly to implement with traditional ballots. For example, offering paper ballots in ten different languages could incur significant costs if the distribution of voter languages is estimated incorrectly. On an electronic system, the only additional cost of providing multilingual ballots is the translation of the ballots (and implementation of the software to display the different ballots).

## Experiences with Electronic Voting

Despite the potential benefits, DREs introduce costs and complexities of their own. Electronic voting machines are expensive; in some US voting districts, costs per voter rose by nearly sixty percent when switching from punch card to fully electronic voting systems [4]. While some of this cost includes the one-time cost of purchasing the hardware, maintenance of the new machines and training of election officials to administer them is a significant expense.

The security implications of the shift from paper to DRE are even more concerning than the cost implications. In any voting system, it is desirable for voters to have the ability to ensure that the result of an election is valid. To satisfy this requirement, there are three important properties: votes should be cast as intended, meaning that a voter's intentions are represented faithfully to the system. Once cast, votes should be recorded as cast, meaning that the system does not tamper with the votes presented to it. Finally, votes should be counted as recorded, meaning that the final tally of votes accurately represents what was recorded [5]. In paper-based systems, the first property is trivially satisfied; voters mark a piece of paper and examine it before it is cast, so they know that the vote is cast properly. Once a paper ballot leaves the voter's hands, however, he must trust an optical scanner to record his vote as cast and the election officials to ensure that the votes are counted as recorded by the scanners.

Direct electronic recording systems currently in use in the United States do not improve upon the security guarantees of older methods. In fact, they are a regression. Consider, for example, the Diebold AccuVote-TS, the most popular DRE voting machine in the US [6]. Since the AccuVote contains closed-source hardware and software, voters have no guarantee of any of the three desired properties. Though the voter is able to see selections they have made on a computer screen, there is no way for them to verify that the votes are actually cast as intended since, for example, options on the user interface may be improperly labeled or otherwise misleading. The voter is also unable to verify that the vote is recorded as cast since software might change the cast vote, either maliciously or due to a software error. Finally, the voter has

no guarantee that the votes are actually counted as recorded, for the same reasons as in traditional voting systems.

One reason for the lack of security in DRE machines like the AccuVote is complexity of the software running on them. Software complexity poses a serious threat to DRE systems, since larger codebases inherently leave more room for errors that affect the outcome of an election. While opening the source code to more extensive public review might help improve its security, DREs' large, complex software stacks make security audits much more difficult. The security issues that are routinely found in large, popular open source projects (e.g., [7]) illustrate this issue. Although it is true that the AccuVote is susceptible to numerous security flaws [6], it also is important to note that the lack of verification provided by current DREs is unrelated to the security of the machine's implementation. Redesigning a machine with more secure, publicly auditable software and better protections against hardware tampering might make voters feel more confident in the results, the improved system would still not provide voters with a true, verifiable guarantee that their votes were recorded and counted properly. To provide such guarantees, a different model is needed.

## Security and Verifiability

There have been some attempts at remedying the verifiability issues with current DREs. The AccuVote-TSx (a successor to the original AccuVote-TS) includes a feature known as a Voter-Verifiable Paper Audit Trail (VVPAT). Using this feature, a voter's selection on the machine's touchscreen is also printed on a piece of paper next to the machine. The voter is given an opportunity to review this paper trail to ensure that the recorded vote is as intended, and then the paper is dropped into a collection barrel, where it can be used in the event of a recount. This system does little to address the threat posed by malicious software, however: if the machine displays the voter's true choice of candidate A on the screen and paper but actually records a vote for candidate B, the discrepancy will only be noticed if the paper trail is recounted by hand (which is unlikely in the majority of elections) [8]. Since the voter has no control over or access to the paper trail after voting, there is no way to independently verify that the vote was cast correctly.

For the amount of money the United States has spent on moving to electronic voting, very little, if anything, has been purchased in the way of security. Current DRE systems have not even succeeded in their original goal of reducing the cost of elections. Electronic voting in its current form is a complete failure. But this does not mean that other electronic voting systems could not be a useful tool for running elections. Current schemes do not take advantage of some of the most compelling applications of computers and software to the voting process. Specifically, they do not incorporate well-known cryptographic techniques that could make the election process verifiable by individual voters. A system that uses these techniques would give voters a true mathematical guarantee that their votes have been recorded properly, instead of forcing them to trust election officials and the manufacturers of voting hardware and software to be honest and competent.

## Approaches

There are two approaches to implementing a more secure electronic voting system. The first is to extend or redesign existing fully electronic (DRE) systems to include the desired security properties. The second approach is to eliminate DRE systems and instead extend paper-based (e.g., optical scan) methods to provide similar security features. This paper begins with a

brief review of the major cryptographic techniques common to both types of system, including homomorphic encryption, zero-knowledge proofs, and threshold decryption. It then considers two systems representative of each technique: VoteBox, a fully electronic DRE system, and Scratch and Vote, a paper-based scheme. Specifically, we will examine how these systems work, and how they provide guarantees about the verifiability properties already discussed. We will also discuss the privacy guarantees associated with each system; privacy plays a role in election security because it prevents vote buying, coercion, and other attacks on the voting process. Finally, the paper will conclude with a discussion of practical aspects of each approach. Based on that discussion, it will argue that the paper-based approach is the most practical way to achieve a return on the investment being made in electronic voting.

## Cryptographic Techniques

It is relatively easy to construct a voting scheme that is verifiable in the sense described above: gather all the voters in a public location and ask them to announce the name of the candidate for whom they wish to vote. Each voter can individually verify the result of such an election, since all voters hear all votes and can keep a tally of their own. Any cheating in the record keeping by election officials would be quickly discovered. Aside from being very impractical for a large electorate, this system does not preserve the secrecy of voters' choices, making it vulnerable to manipulation by those who would bribe or coerce voters into choosing a particular candidate. To prevent manipulation, voters' choices must be kept a secret from others. More subtly, voters must also be unable to prove how they voted to anyone else, because this would also allow bribery and coercion. The aim of cryptographic techniques described in this section is to enable a voting scheme to have the verifiability of a 'public square' system while retaining the privacy and practicality of a traditional secret ballot. It is assumed that the reader has a basic knowledge of asymmetric cryptography and modular arithmetic.

### Homomorphic Encryption

Homomorphic encryption is a technique used by almost all of the systems considered in the following sections. A ciphertext generated by a homomorphic encryption algorithm has a property called malleability, meaning that it is possible to modify the ciphertext to produce a second ciphertext whose decryption is related to the original's plaintext [9]. Malleability is typically considered a negative property of a cryptosystem, since it might allow an attacker to modify encrypted messages sent over an insecure channel in malicious ways. Homomorphic systems use this property to their advantage, however, using malleability to enable computation (typically addition or multiplication) over encrypted data without revealing the original information. This is useful for voting systems because it allows voters' selections to be published in an encrypted, unreadable form, while still allowing those votes to be counted. When the final tally in decrypted, no information about individual voters' choices is revealed, thus preserving the secrecy of the votes.

A homomorphic encryption scheme can be fully or partially homomorphic. Fully homomorphic schemes allow multiple operations to be performed on encrypted data (e.g., addition and multiplication), whereas partially homomorphic systems only allow one. Although fully homomorphic schemes do exist [10], they are substantially more complicated than partially homomorphic schemes and are not particularly useful for voting.

One partially homomorphic system is the Pallier cryptosystem. The full mathematical details are beyond the scope of this paper, but a few key points follow (more details can be found in [11]). During encryption, plaintexts are multiplied by a random number modulo $n$, where $n$ is a public parameter of the system. This prevents attackers from generating a table of possible ciphertexts in situations where the space of plaintexts is relatively small. In an election, for example, a voter might only need to encrypt a single bit signifying a vote for or against a candidate. Without this randomizing factor, it would be trivial for anyone with access to the public encryption key to determine the voter's selection. During the decryption operation, the random term falls out, leaving the original plaintext. The important homomorphic property of this system is that multiplication of ciphertexts modulo $n^2$ results in a ciphertext that represents the sum of the plaintexts. In other words, if $E$ and $D$ are the encryption and decryption functions, respectively, then $D\left(E_{r_1}(m_1) \cdot E_{r_2}(m_2) \; mod \; n^2\right) = \; m_1 + m_2 \; mod \; n$. Note that this property only holds when the ciphertexts being combined are generated using the same public encryption key. It is not possible for two parties to encrypt secrets using separate public/private key pairs, combine the result, and then somehow recover the sum of those secrets. Another homomorphic system, a variant of ElGamal, is also used by some voting systems, one of which is described later. This ElGamal variant has similar characteristics as Pallier encryption and can be considered equivalent for the purposes of this paper.

Homomorphic encryption is useful for tallying votes while preserving privacy, but it suffers from a security flaw when used for that purpose. Consider a scenario in which voters encrypt their choices using a homomorphic system, sending them to some central authority which tallies the votes. Suppose that a vote of '0' indicates a vote for candidate A, while a vote of '1' indicates a vote for candidate B. The votes are added using the homomorphic property of the cryptosystem; if the tally is greater than half the total number of voters, candidate B wins. Otherwise, candidate A wins. What happens if a voter encrypts the value '100' as their vote? In this example, the voter has entered 100 votes for candidate B. Since the value is encrypted, nobody is able to inspect the vote to ensure that it contains a valid value (i.e., a '0' or '1'). As we will see, actual voting systems use homomorphic encryption in a more sophisticated way than this example, but the general vulnerability remains the same. To avoid this problem, a technique called a zero-knowledge proof is required.

## Zero-Knowledge Proofs

A zero-knowledge proof is a technique for proving a statement while revealing nothing (i.e., zero knowledge) about the statement in question. The following example (roughly adapted from [12]) clarifies this concept: Alice is selling a dictionary that she wrote. The dictionary contains extremely helpful definitions, but unfortunately is published in non-alphabetical (random) order and without an index. Alice sells the index for an additional fee. Bob would like to buy the dictionary, but before he pays the outrageous fee Alice is charging for the dictionary, he'd like to make sure that she actually has an index for it. Alice wants to prove to Bob that she has the index, but she also doesn't want to reveal its contents, since then Bob would have no reason to buy it later. They agree to the following protocol. Bob names any word in the dictionary, and Alice quickly (e.g., within 30 seconds) turns to the page in the dictionary containing that word. Alice obscures everything on the page except the requested word and covers the dictionary with a special cover that makes it impossible to see how far into the book she has turned. She then allows Bob to inspect the page to ensure that the word he requested is shown. They repeat this procedure many times (perhaps 100). If Alice ever fails to find the

correct page, Bob can assume that she was lying about having an index. If Alice never fails, however, it is likely that she does have an index, since the odds of her guessing the correct page for 100 words are very slim. Bob has proof that Alice has an index, but he has not learned the page number or definition for any of the words in the dictionary. So Alice has given a zero-knowledge proof to Bob that she has the index.

In the context of voting, zero-knowledge proofs can be used to prove that encrypted votes have certain characteristics without revealing the votes themselves. For example, a zero-knowledge proof attached to a homomorphically encrypted vote might prove that the vote plaintext is either a '0'or '1'. This prevents the voter in the previous example from submitting a vote of '100', but also does not reveal any information about the actual vote. In practice it would be impractical for the back-and-forth communication of Alice and Bob to occur for every vote in a system. The voting systems considered later use a variant known as non-interactive zero-knowledge (NIZK) proofs, which allow the proof to take place with no interaction between the prover and verifier, provided that the parties have access to a small amount of shared data [13]. This allows votes to be verified by election officials and other observers without requiring interaction with each voter.

## Counting Encrypted Votes

Using homomorphic encryption and NIZK proofs, encrypted votes can be cast by in the open, without loss of voter privacy and without fear of invalid votes being cast. Eventually, however, the final tally must be decrypted to learn the results of the election. As noted earlier, homomorphically encrypted votes must be encrypted using a single public key for the combination of the votes (i.e., tallying) to work. This implies that all of the votes can also be decrypted by a single private key. The holder of this key not only has the power to decrypt the final tally of votes, but also individuals vote. This allows the holder to single-handedly bypass the system of voter privacy that the previous techniques are designed to establish. In addition to the threat of corruption, a single key holder could potentially lose the private key, for example due to the loss of a data center. This would be extremely costly, as election results would be unrecoverable and the election would need to be run again. In voting systems, a technique called threshold encryption is used to prevent such a single point of failure.

Threshold encryption is based on a technique developed by Shamir [14], which allows a secret number to be divided into $n$ shares, at least $k \leq n$ of which are required to recover the secret. The technique is based on the observation that a degree $k-1$ polynomial is uniquely defined by $k$ or more points that lie on its curve. A polynomial $q$ is generated such that its $0^{th}$ degree coefficient is the secret number, and shares of the secret are generated by evaluating the polynomial at $n$ distinct points. When $k$ or more of these shares are combined, the original polynomial can be derived, and the secret can be obtained by evaluating $q(0)$. This method can be applied in a straightforward way to distribute private key material amongst several stakeholders. In particular, it can be applied to the Pallier cryptosystem described above [15]. Threshold Pallier encryption makes it impossible for a single election official to decrypt votes (or lose the only decryption key). Only decryptions that are authorized by a sufficient number of stakeholders may proceed. In an election, these stakeholders might be representatives of major political groups, election watchdog groups, or other government agencies. It is unlikely that a majority of these stakeholders would agree to decrypt individual votes. Assuming this is true, nothing but the final tally is ever decrypted.

We have seen how several cryptographic methods can be used together to enable public broadcasting of private, yet provably valid, votes, and how these votes can be tallied and decrypted by a group of trustworthy parties. In the following sections, we examine how these basic techniques can be combined in various ways to form more complete voting systems.

## Fully Electronic Schemes

The two approaches to solving the security issues of current DRE systems are to design improved DRE systems or develop new paper-based schemes. Here we examine one proposal for an improved fully electronic (DRE) system called VoteBox [16]. The VoteBox scheme aims to solve the security issues of DREs both by reducing the amount of software in voting machines that must be trusted to operate correctly, and by fundamentally improving the DRE voting model.

### Managing Complexity

VoteBox significantly reduces the amount of trusted code in the system by forcing the user interface presented by the machines to be designed and rendered prior to the election. User interface rendering code constitutes a significant fraction of the total codebase for a traditional DRE. In the AccuVote-TS, for example, UI software makes up almost half of the total code (measured by lines) running on the machine [17]. In a pre-rendered UI, interface elements such as buttons and text are not drawn in real-time by a user interface framework. Instead, each view presented to the voter is a single, static image that is drawn to the screen, along with some information defining regions that trigger actions. When an action requires a new view to be displayed, an entirely new image is drawn to screen. This reduces the complexity of the state that must be tracked by the voting machine: rather than text positioning information, toggle states of buttons, etc., the machine keeps track of the current image being displayed. In addition to reducing the amount of code that must be verified, pre-rendered interfaces have the advantage that the exact layout and text of each ballot can be examined prior to the election, without fear that a software issue will make it unusable.

### Voting Setup and Procedure

Unfortunately, there is only so much code that can be removed from a DRE. After removing as much as possible, VoteBox's design aims to make the remaining software as easily verifiable and robust against failure as possible. In the event that there is a software flaw, this verifiability enables misbehaving machines to be easily identified and removed from polling stations, and also that voting information is not lost. One of the key components of VoteBox that makes this possible is a logging system known as the Auditorium. During an election using VoteBox, machines in each polling place are connected to a local network. As the election takes place, every action taken by any machine is broadcast in a message to the Auditorium network and is logged by all other machines. Each of these messages is cryptographically signed by the originating system. Additionally, each machine keeps a running hash of all messages in the Auditorium record, known as a hash-chain. When messages are broadcast, they are accompanied by a hash of the current message along with the previous hash in the chain. This makes it very difficult for an individual machine to present an inaccurate record of events, since modifying a message in its log causes the hashes of all subsequent messages to be incorrect (since all other machines saw the original message, and incorporated its value into their hash chains). In

addition to being hard to tamper with, the broadcast property of the Auditorium log makes it durable against the failure of multiple voting machines, since each machine has a complete copy of the log. So unless all machines in a polling place's network fail or collude to corrupt the election, a correct log of the actions of every machine will remain intact.

When an election begins, a poll worker uses a special supervisor machine, also connected to the Auditorium network, to instruct the voting machines at each booth to begin accepting votes. To do this, the worker enters a secret random number, provided by an election authority, into the supervisor machine. This number is logged in the Auditorium as part of the message from the supervisor machine signaling the polls' opening, and serves to verifiably mark the beginning of the election. Any log messages that occur after the secret number must have come sometime after it was distributed to polling places. After the polls are open, voters interact with the booth machines to make their selections. When votes are cast, they are encrypted using an ElGamal-derived homomorphic cryptosystem, which has a similar additive homomorphic property as the Pallier system described earlier. Like the Pallier system, votes are combined with a random number to keep the ciphertext from being easily identifiable. After encryption, votes are broadcast to the Auditorium log. VoteBox does not attach a NIZK proof to the encrypted value. So, the Auditorium log generated by voting activity serves as a tamper-proof, durable record of all votes cast. The voter, however, has no guarantee that this record accurately represents the ballot cast, since he has no guarantee that the encrypted vote broadcast to the Auditorium is for the correct candidate, or that it even encodes a valid number of votes.

## Challenge-based Verification

To give voters a guarantee that polling machines are operating correctly, VoteBox implements a challenge protocol. Once a voter completes a ballot and confirms the choices made, an encrypted vote is broadcast to the Auditorium. The voter is then presented with a screen asking them to either finalize the vote, in which case a finalization log is written, or to challenge the machine to prove that it is operating correctly. If the voter elects to challenge, the machine is required to broadcast an additional message containing the random number used to encrypt the vote plaintext, effectively revealing the contents of the vote. After this point, the vote is considered invalid and will not be counted when the logs are reviewed, since the vote is no longer anonymous. To actually carry out the verification, VoteBox relies on a final property of the Auditorium system: it is connected via a one-way device known as a data diode (described in [18]) to the Internet. Every log message broadcast on the network within a polling station is also reported in a publicly accessible feed. It is assumed that external voting watchdog groups will monitor these streams and do the necessary cryptography using the encrypted vote and random value published by the machine to verify to the voter that the ballot was encrypted correctly. It is important that the encrypted vote be broadcast on the auditorium before the voter chooses to challenge (otherwise the machine could always make sure the behave correctly in a challenge situation). So a voter who wishes to challenge a machine will first notify a verifying organization, who would likely be set up in the polling place. The voter will make their selections and wait until they are notified by the organization that an encrypted ballot as been published. The voter will then select the challenge option instead of choosing to finalize the vote, and will confirm with the organization that the encrypted ballot turned out to be correctly generated.

Importantly, VoteBox's challenge system does not actually allow a voter to verify that a finalized ballot is correctly encrypted. However, since a machine must always publish the

encrypted ballot before it knows if it will be challenged, it cannot behave incorrectly without the risk of being challenged and caught. Assuming that enough systems are challenged, it is unlikely that a misbehaving system would escape detection. So the challenge system provides voters with a (almost) guarantee that their votes are cast as intended and recorded as cast, and the Auditorium logs prevent the record from being lost or tampered with. All logs are public, so it is also possible to verify that the votes are counted as recorded, since all vote information is available in the logs, and any individual can perform the homomorphic operations to tally the vote (although this tally can only be decrypted once the private key material is released).

## Paper Based Systems

Having considered a fully electronic proposal, we now examine a paper-based alternative to current DREs. Rather than attempt to improve electronic systems, paper-based systems build on traditional optical scan technology by augmenting paper ballots with cryptographic technology. The system considered here is called Scratch and Vote [19], so named for the scratch-off portions included on its ballots. There are many paper-based schemes, each varying in its ballot design and use of cryptography to secure votes. Scratch and Vote is a good representative system because it is relatively simple while still using all of the previously described cryptographic techniques. Additionally, it is conceptually similar to VoteBox, making it a good system to consider when comparing the electronic and paper-based approaches to voting.

### Ballot Design

The Scratch and Vote ballot has three sections, separated by perforations in the paper that allow them to detached. The first section contains the names of the candidates in the race listed in a random order on each ballot (i.e., two ballots in the same race may have different candidate orderings). Most elections involve more than a single race. For convenience, Scratch and Vote ballots for multiple races can be combined on a single page, as long as the three components for each race can be separated from the other ballots.

The second ballot section contains a series of computer-readable selection areas (e.g., bubbles, checkboxes, etc.), similar to a normal optical-scan ballot. It also includes a barcode containing the homomorphically encrypted votes corresponding to each bubble and a hash of the public key used to generate the ciphertexts. The votes are encrypted using the Pallier cryptosystem. In order to support more than two candidates, a vote is represented by a single number whose bits are partitioned into a region for each candidate. For example, an election with three candidates and 1000 voters might represent votes as 30 bit numbers, with ten bits for each candidate. The design of the ballot is flexible, but the first and second portions of the ballot should typically be arranged so that it is obvious to the voter which candidate is associated with each bubble.

The third section of the ballot contains a barcode covered by an opaque scratch-off material, as is commonly found on lottery tickets or gift-cards. This barcode contains the random values used during the encryption of the plaintext votes (recall that the Pallier system encrypts plaintexts with an additional random value in order to prevent ciphertext-lookup attacks). These values are hidden because, combined with the public key, they are sufficient to decipher the encrypted votes stored on the barcode. Except when it is used for verification purposes (described below), this portion of the ballot is discarded without being revealed.

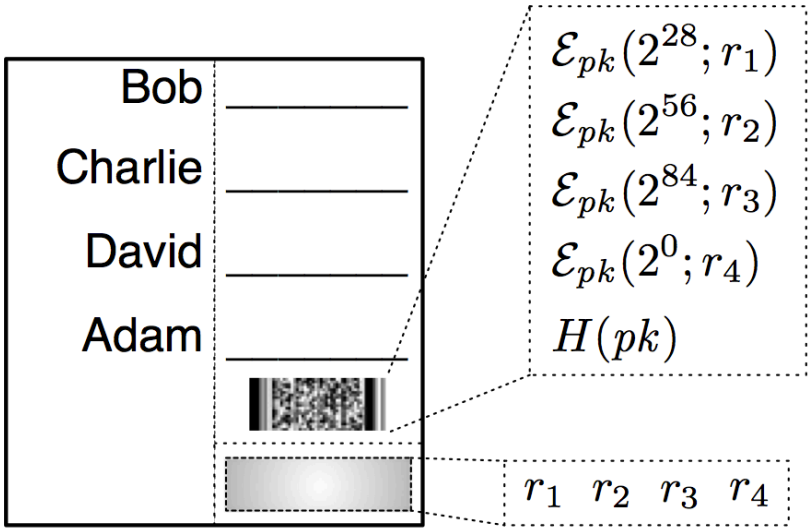


**Figure 1: Illustration of the Scratch and Vote Ballot. Source: [19]**

In addition to the elements printed on the ballot itself, a series of NIZK proofs is generated for each ballot and posted on a public 'bulletin board' website prior to the election. These proofs guarantee that the encrypted votes on the corresponding ballot are valid. During and after the election, this public bulletin board is used to post additional information that aids in the verification process.

## Voting Procedure

To vote, a voter enters a polling place and receives a paper ballot. The voter makes his choices by filling in the bubbles corresponding to the desired candidates. He then removes the list of candidate names and discards it. After the list of candidates is discarded, the vote is irrecoverable without the information hidden behind the scratch-off surface or the private encryption key. The position of the marked area reveals nothing about the contents of the vote, since the candidates were listed in a random order. The voter hands the ballot to an election worker, who ensures that the scratch-off surface on the third section of the ballot is intact. The worker then separates the third section and discards it. After this step, the vote cannot be recovered without the private key. The remaining portion of the ballot is scanned by an optical scanner, which records the selection and the information stored on the barcode. The scanner also captures a complete image of the ballot, which is posted on the public bulletin board. The ballot image can be associated with the name of the voter so that it is easy to retrieve. The voter receives the original paper copy as a receipt.

Like VoteBox, Scratch and Vote allows voters to carry out an optional verification step while at the polling place. Though the NIZK proofs for each ballot ensure that the encrypted votes associated with each selection only record a single vote, they do not guarantee that the ciphertexts correspond with the candidate labels printed on the ballot. For example, it is conceivable that the candidate list in the first section of the ballot is permuted differently than the (encrypted) vote list encoded on the second section's barcode. A voter who wishes to perform a verification step obtains two ballots when entering the polls. The voter selects one of the ballots

and scratches off the surface on the third ballot section to the randomization values. The voter then approaches a third-party watchdog group. This organization scans the barcode on the second ballot section and the randomization values on the third section, computes the votes associated with each selection region, and verifies that they match the list of candidates on the first section. Assuming that the verified ballot is valid, the voter proceeds to vote with the second ballot. The verified ballot is discarded: it is not considered valid because the votes are not anonymous once the randomization values have been revealed. As with VoteBox, this verification process does not actually provide proof to a voter that the ballot they used to vote was valid. The assumption is that, with enough verifications, widespread tampering with ballot ordering will be detected.

### Post-election Verification

Verification at the polling place gives voters confidence that ballots are cast as intended, i.e., the marks that they placed on ballots corresponded correctly to the desired candidates. Post-election verification allows voters to ensure that their votes were recorded as cast and counted as recorded. Since all ballot images are posted on a public bulletin board, individual voters can go online and verify that the ballot image associated with their name matches the receipt they took out of the polling place. Individual voters or watchdog organizations can also independently perform the tallying process on the homomorphically encrypted votes, ensuring that all votes pictured on bulletin board make their way into the final tally. They can also use the NIZK proofs associated with each ballot to ensure that the values being tallied each only represent a single vote. Once election officials perform their tally, they post it to the bulletin board so that verifiers can ensure that the value matches the independently calculated tallies.

In addition to ensuring a correct tally of the posted ballots, organizations with access to voter registration information can also verify that every ballot included on the bulletin board (and therefore included in the final tally) corresponds to a registered voter, and that each registered voter has only one associated ballot. This prevents vote-stuffing attacks, since the origin of every ballot is known.

## Discussion

It is clear that both of the techniques discussed above, when implemented properly, provide substantially more security than existing DRE or paper-based systems. From a technical perspective, VoteBox and Scratch and Vote use similar technologies provide nearly identical verifiability and privacy guarantees. When considering which approach to adopt, however, it is important to also consider the practical aspects of each technology. These include the cost and complexity, accessibility to voters, and political feasibility of each approach.

### Cost and Complexity

The VoteBox system requires a substantially more setup at each polling place than Scratch and Vote. The most obvious of these requirements is a voting machine at each voting booth. In contrast, paper-based schemes like Scratch and Vote only require enough optical scanners to scan each voter's ballot as they exit the polling place. Since the actual selection of candidates is likely the most time-consuming aspect of each voter's experience, this suggests that this number is far smaller than the number of voting machines required by VoteBox. Current DRE voting machines are expensive: in one state, a 2005 procurement contract lists the purchase

price of an AccuVote-TSx voting machine at over 3000 dollars [20]. While it is likely that the cost of voting machines used in a DRE system could be lowered significantly, the cost of such machines will certainly not become negligible.

In addition to the voting machine requirements, the networking required by VoteBox's Auditorium system adds significant complexity to polling place setup and administration. Each VoteBox machine must be able to communicate with every other machine in the polling place. This requires additional hardware in the form of network switches, as well as the specialized data-diode device that establishes a one-way connection from the polling place to the rest of the world. These devices not only add cost to each polling place, they also introduce additional complexity into poll place setup. In 2012, nearly one quarter of all poll workers were 71 years or older, with an additional sixty percent aged between 40 and 70 years old. In some states, the majority of poll workers were over 60 [2]. Given the age distribution of workers, it is unlikely that many will have the skills necessary to properly network the machines prior to Election Day or troubleshoot issues as they arise (e.g., a switch or router failure). Such networking failures could significantly impact the voting process at affected polling places, since voting (and challenge-based verification) cannot proceed without a functioning Auditorium network. In contrast, Scratch and Vote does not require any networking capability. It would be convenient to be able to directly upload scanned ballots to the public bulletin board, but this is not strictly necessary, since votes can always be transferred to a central location to be uploaded after the fact.

Outside of the polling place, both systems introduce additional costs and complexities to the election process. Both require a public bulletin board system that is capable of storing and presenting election data to the public in a secure fashion. The software used to upload information and run the bulletin board itself will be expensive to produce and administer, especially considering the high volume of traffic that it will receive. Failure of the bulletin board system on Election Day due to heavy load would be an unacceptable failure that would jeopardize the election. In addition to bulletin board software, both systems will require a well-written, publicly audited cryptography library to implement the necessary encryption and verification functions. Finally, in the case of the VoteBox system, additional software is required to actually run the voting machines. In addition to the one time cost of producing such software, there will be additional costs to maintain and update it as necessary.

In addition to software requirements, Scratch and Vote also requires that its ballots be protected and kept a secret prior to the election. Once a complete ballot is revealed, it is no longer anonymous, since the ordering of the candidates is revealed. An attacker with the ability to photograph or scan a large number of intact ballots could compromise the secrecy of the votes cast on the captured ballots. Due to this risk, the paper-ballots used by Scratch and Vote must be protected from wherever they are printed all the way to hands of the voters. One reasonable approach to the problem is to wrap ballots in an opaque, tamper-evident seal after they are printed. This would limit the ballots' vulnerability to the time between when they are printed and when they are sealed, which could conceivably be made quite short. This scheme would also increase the cost of the Scratch and Vote ballots relative to traditional paper ballots, however. In addition to the protective seals, the scratch-off portion on every ballot would add additional cost and complexity to ballot production.

It is difficult to quantify the costs associated with each of the factors discussed in this section. VoteBox requires expensive voting machines, but Scratch and Vote requires ballots that are complicated to produce and protect; both require significant software development (Scratch

and Vote requiring somewhat less), whose cost is notoriously difficult to predict; paper-based systems miss out on savings that DREs gain by avoiding additional costs for ballots in different languages or formats. Producing an accurate estimate of how these costs will balance out is beyond the scope of this paper (although the data introduced in the discussion suggest that DRE systems raise costs substantially). Cost aside, however, VoteBox introduces far greater complexity into the polling place than Scratch and Vote, in particular due to the requirements of the Auditorium network. It may even be worth a slightly higher cost to avoid the high risk of procedural issues with such a system.

## Voter Training

Another important consideration for new voting systems, perhaps more than cost and complexity, is how well the average voter understands and trusts each system. If voters do not understand how to use a system, they will be unable to vote properly. If they do not believe that a system works, they will be unwilling to use the system at all. Given the distribution of voting technologies discussed earlier, it is likely that many voters have at some time used either a DRE or a paper-based optical scan system. On the surface, VoteBox and Scratch and Vote are just normal DRE and optical scan systems, respectively, so it is unlikely that voters would be confused about how to cast a ballot. Both systems rely on an additional verification mechanism, however, which has the potential to confuse voters.

Voter participation in verification steps is a crucial component of both systems' security. Both rely on the assumption that enough voters will challenge ballots or machines that the probability of undetected tampering is sufficiently low. In the case of VoteBox, voters who wish to challenge a machine must go through a somewhat complicated procedure in which the alternate between communicating with a helper organization and interacting with the machine (recall that the voter must inform a helper organization before actually telling the machine that they would like to challenge, in order to ensure that the machine does not cheat). On the other hand, a Scratch and Vote ballot challenge simply requires that the voter bring a ballot to a helper organization. The ballot is then verified in front of the voter in a single step. Explaining the rationale behind the additional steps required by the VoteBox challenge protocol to a non-technical voter might be difficult: it requires describing the sequence of messages sent by the machine, and then explaining why failing to communicate with the helper organization between confirming ballot choices and challenging the machine could allow the machine to disguise malicious behavior. The process of verifying a Scratch and Vote ballot is non-interactive, which might be more intuitive to the average voter: the ballot contains some hidden information that validates its contents, which the helper organization checks.

Regardless of which system's verification method is easiest to explain, what is ultimately most important is how likely voters are to actually perform verifications/challenges under each system. Scratch and Vote's method allows the voter to verify a ballot either before or after they vote, and only requires a single interaction with the helper organization. VoteBox requires multiple interactions at specific times in the voting process. The added hassle of the VoteBox procedure will likely discourage voters from performing challenges, which could reduce the system's security. User studies are necessary to definitively decide which system is easiest to explain and which system voters are most likely to use, but the simplicity of the Scratch and Vote system is almost surely an advantage.

A final aspect of each system's voter experience worth considering is how easily the system can be adapted to support assistive technologies and alternate languages. This is an

important consideration: census data indicates that close to five percent of the US population speaks English "not well" or "not at all" [21]. A key advantage of DRE systems in general is that they allow ballots to be easily adapted into alternate languages. This is true of the VoteBox system, which has the added advantage that the pre-generated ballots can be checked beforehand in order to guarantee that translations are correct and will render properly on screen. Scratch and Vote ballots can be translated but, like all paper systems, suffers from the weakness that translations add cost if the distribution of languages is not predicted correctly. In addition to translations, VoteBox also has an advantage in the area of accessibility, since electronic systems tend to lend themselves to (usually electronic) assistive devices, whereas paper systems require either human assistance or costly special ballots.

## Other Issues

We now close with a brief discussion of other factors that may affect the adoption of improved voting systems. One such factor is election law in different states, which may unintentionally prohibit certain aspects of each system. For example, different states stipulate the ordering of candidates on ballots in very particular ways, the goal being to minimize the effect of ordering on the election outcome [22]. This type of law may conflict with Scratch and Vote's requirement that the order of candidates is listed in a different random order on each ballot. While in theory this is the fairest way of ordering candidates, it may run astray of laws that require every ballot to have the same random order. Changing the law to allow Scratch and Vote should be an easy legislative fix, but will require political will at the state or national level in order to actually change. This means that individual municipalities that wish to experiment with new systems like Scratch and Vote may have a difficult time doing so until the laws are changed. Similarly, one of the challenges discussed in the original VoteBox paper are laws that prohibit connecting voting equipment to networks. Such laws would clearly prevent the VoteBox protocol from functioning, and would need to be changed before it could be implemented.

Finally, it is worth considering how preferential voting might be implemented on either system. Although the US does not currently use a preferential voting system, it is common in many places. Should the US decide to adopt such a system sometime in the future, it would be helpful to have adopted a voting system that can be adapted accordingly. A simple solution to the preferential voting problem is to encode each permutation of candidates as a single choice. This quickly becomes unmanageable, however, since the number of permutations grows rapidly. Another potential solution is to simply count each preference position (e.g., first choice, second choice, etc.) as a different race, combining the results from each 'positional' race to calculate a final result. Unfortunately, this might have unforeseen results on privacy guarantees. More advanced techniques, beyond the scope of this paper, have been developed to allow preferential voting using homomorphic counters [23]. Using these advanced techniques, both systems should be adaptable to preferential voting.

In this paper, we have examined the motivation for developing an improved, cryptographically enhanced voting system, focusing on two potential alternatives: fully electronic and paper based. After examining three important cryptographic techniques that make such systems possible, we discussed two systems representative of the two approaches. Based on these examinations, we found that both approaches offer similar guarantees of security and privacy. However, we also found that VoteBox, the fully electronic scheme we considered, adds

significant costs and complexities over a paper-based scheme. Furthermore, it is potentially more difficult to explain to voters, and more difficult to get voters to participate in elements of the protocol that are essential security. Looking at legislative issues, both systems will require changes to the law. Given the fact that both VoteBox and Scratch and Vote provide similar security and privacy guarantees, and the fact that electronic schemes like VoteBox add additional complexities and costs, it makes sense to invest in paper-based voting systems as the next generation of voting technology.

## Works Cited


[1] "A Summary of the 2004 Election Day Survey," United States Election Assistance Commission, 2005.

[2] "2012 Election Administration and Voting Survey: A Summary of Key Findings," United States Election Assistance Commission, 2013.

[3] Vir Singh. (2004, May) IEEE Spectrum. [Online]. http://spectrum.ieee.org/computing/software/electronic-voting-eases-india-elections

[4] SAVEOurVotes. (2008, February) Cost Analysis of Maryland's Electronic Voting System. [Online]. http://www.saveourvotes.org/legislation/packet/08-costs-mdvotingsystem.pdf

[5] Ben Adida and C. Andrew Neff, "Ballot Casting Assurance," in *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop*, Berkeley, CA, 2006, pp. 7-7.

[6] Ariel J Feldman, J. Alex Halderman, and Edward W Felten, "Security Analysis of the Diebold AccuVote-TS Voting Machine," in *Proceedings of the 2007 USENIX/ACCURATE Electronic Voting Technology Workshop*, 2007.

[7] Dan Goodin. (2014, March) Critical crypto bug leaves Linux, hundreds of apps open to eavesdropping. [Online]. http://arstechnica.com/security/2014/03/critical-crypto-bug/

[8] Joseph A. Calandrino et al., "Source Code Review of the Diebold Voting System," Office of the California Secretary of State, 2007. [Online]. http://www.sos.ca.gov/voting-systems/oversight/ttbr/diebold-source-public-jul29.pdf

[9] D. Dolev, C. Dwork, and M. Naor, "Nonmalleble Cryptography," *SIAM Journal on Computing*, vol. 2, no. 30, pp. 391-437, 2000.

[10] Craig Gentry, "A fully homomorphic encryption scheme," Stanford University, Palo Alto, CA, PhD Thesis 2009.

[11] Pascal Pallier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, 1999, pp. 223-238.

[12] Jean-Jacques Quisquater, Louis Guillou, Marie Annick, and Tom Berson, "How to explain zero-knowledge protocols to your children," in *Proceedings on Advances in cryptology*, Santa Barbara, CA, USA, 1989, pp. 628-631.

[13] Uriel Feige, Dror Lapidot, and Adi Shamir, "Multiple noninteractive zero knowledge proofs under general assumptions," *SIAM Journal on Computing*, vol. 29, no. 1, pp. 1-28, 1999.

[14] Adi Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, November 1979.

[15] Pierre-Alain Fouque, Guillaume Poupard, and Jacques Stern, "Sharing Decryption in the Context of Voting or Lotteries," in *Proceedings of the 4th International Conference on Financial Cryptography*, London, UK, 2000, pp. 90-104.

[16] Daniel R. Sandler, Kyle Derr, and Dan S. Wallach, "VoteBox: A tamper-evident, verifiable electronic voting system," in *Proceedings of the 17th USENIX Security Symposium*, 2008.

[17] Ka-Ping Yee, David Wagner, Marti Hearst, and Steven Bellovin. (2006, April) Prerendered User Interfaces for Higher-Assurance Electronic Voting. [Online]. http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-35.html

[18] D. W. Jones and T. C. Bowersox, "Secure data export and auditing using data diodes," in

*Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop*, Vancouver, 2006.

[19] Ben Adida and Ronald L. Rivest, "Scratch & vote: self-contained paper-based cryptographic voting," in *Proceedings of the 5th ACM workshop on Privacy in electronic society*, New York, NY, 2006, pp. 29-40.

[20] Texas Comptroller of Public Accounts. (2005, March) Contract Terms and Instructions: 578-N1-ELECTRONIC VOTING SYSTEMS. [Online]. http://www.window.state.tx.us/procurement/contracts/gci/578-N1.php

[21] Camille Ryan. (2013, August) Language Use in the United States: 2011. [Online]. http://www.census.gov/prod/2013pubs/acs-22.pdf

[22] R. Michael Alvarez, Sinclair Betsy, and Richard L. Hasen, "How Much Is Enough? The "Ballot Order Effec" and the Use of Social Science Research in Election Law Disputes," *Election Law Journal*, vol. 5, no. 1, pp. 40-56, 2006.

[23] Eu-Jin Goh and Philippe Golle. (2005) Event Driven Private Counters. [Online]. http://crypto.stanford.edu/~pgolle/papers/preferential.pdf