Attacking Paper-Based E2E Voting Systems

John Kelsey[1], Andrew Regenscheid[1], Tal Moran[2], and David Chaum[3]

[1] National Institute of Standards and Technology {john.kelsey, andrew.regenscheid}@nist.gov
[2] Weizfmann Institute of Science tal.moran@weizmann.ac.il
[3] info@chaum.com

Abstract. In this paper, we develop methods for constructing vote-buying/coercion attacks on end-to-end voting systems, and describe vote-buying/coercion attacks on three proposed end-to-end voting systems: Punchscan, Prêt-à-voter , and ThreeBallot. We also demonstrate a different attack on Punchscan, which could permit corrupt election officials to change votes without detection in some cases. Additionally, we consider some generic attacks on end-to-end voting systems.

1 Introduction

Voting systems in widespread use today have a number of known vulnerabilities [1–3]. Many of these vulnerabilities can be mitigated by following certain procedures; the integrity of the election is then dependent on a combination of correct behavior by software, hardware, and election officials.

The best of these systems provide security assurance based on the honesty and correct behavior of a small set of election officials and other observers. Commonly, each political party or candidate provides a certain number of observers. These individuals are expected to notice and report fraud that would deprive their party or candidate of votes. Election officials are also expected to notice and report fraud. In general, an outsider attempting to decide whether to trust a reported election outcome must rely on the premise that correct procedures were followed by observers and election officials.

A new kind of voting system has been proposed in recent years [4–9], in which the voter interacts with the voting system to get a *receipt*. This receipt can then be checked against a set of receipts published by the voting system. These published receipts can be used to produce or verify the reported count for the voting system, but must not be useful for selling votes (for example, by proving how each voter voted). This class of voting system has been called *end to end*, or *E2E*, to reflect the idea that each voter can check, to some high degree of confidence, that his vote was correctly cast, and also that his vote was correctly included in the final count. Other means must be used to ensure that the whole election result is correct- for example, by ensuring that only authorized people were permitted to vote, and that no additional votes were inserted into the count. These systems build on older work on cryptographic voting systems

[10–12], which typically relied on the assumption that a voter would have access to some trusted computing devices.

In this paper, we consider the security of a number of proposed E2E voting systems against attacks to tamper with election results and to permit the buying or coercion of votes.

An important idea in this paper is that most E2E systems can be meaningfully divided into:

1. A *front-end*, which describes the voter's direct interaction with the voting system to cast a vote and receive a receipt, and
2. A *back-end*, which describes the voting system's public statements (such as receipts posted to a bulletin board and the claimed vote totals), and the mechanisms used to prove to voters and other observers that the reported election results are consistent with the public statements.

Our attacks focus exclusively on the front-end, the voting systems' interactions with the voters. In general, our attacks provide ways in which corrupt election officials or voters can subvert the real-world implementation of the front-ends of these voting systems, to get very different properties from the voting systems than were expected.

1.1 Attacking Voting Systems

In general, someone attacking a voting system wants to affect the outcome of the election. The stakes in such an attack can be quite high, involving control of enormous government resources. These stakes can be inferred by the amount of money spent on lobbying and campaigning, both reported publicly in the United States [13, 14].

Election results can be changed by altering recorded votes or reported totals, and also by finding a way to learn what each voter chose, so that voters can be bribed or coerced into voting in some desired way. It can also be done by disrupting the orderly operation of an election, which may simply delay an undesirable (to the attacker) result, or may force an election to be rerun, possibly changing its result. Violations of voter privacy and disruption of elections may be of some interest to attackers even when the election result cannot be altered, but the impact of these attacks is much smaller.

1.2 Previous Work

Several end-to-end cryptographic voting protocols have been developed. This paper analyzes Punchscan [6, 7], Prêt-à-voter [15] and ThreeBallot [8, 16], which are described in Section 2.

Researchers have begun to perform security analyses of these schemes, and some weaknesses have been discovered. A coercion attack against ThreeBallot, dubbed the ThreePattern attack [8], involves a coercer telling voters to mark their three ballots according to a particular pattern, then checking that that those patterns appear on the bulletin board. Strauss [17, 18] notes several vulnerabilities in ThreeBallot, including the Reconstruction attack. Here, for sufficiently long ballots, an attacker is able to look at one receipt and determine which other two ballots on the bulletin board belong to the same multiballot.

A well known issue with these systems is that it is easy to force a voter to vote for a random candidate by instructing them to return with a receipt showing a vote for the first ballot choice. A formal study of the three voting schemes by Clark, Essex and Adams [19] concludes that while ThreeBallot receipts provide some clues for how voters voted, Prêt-à-voter and Punchscan receipts do not contain any information that would help an attacker. Nonetheless, Moran and Naor [20] developed a coercion attack against Punchscan that relied on the voter's choice of receipt. The Punchscan voting procedure was modified for the VoComp competition [21] to prevent this and related attacks by requiring voters to choose the receipt sheet prior to viewing the ballot.

1.3 Our Results

Briefly, our results can be summarized as follows:

Punchscan and Prêt-à-voter

- We provide a misprinting attack which can alter election results by misleading many voters into believing they have a receipt committing to a different vote than is actually cast. This can be mitigated with a special audit of the printed ballots, but that auditing requires trusting small numbers of election observers, rather than all voters, with the integrity of the election.
- We provide a mechanism for using scratch-off cards, cellphones, or other techniques to reliably buy or coerce votes.

Prêt-à-voter

- We report a previously-known but unpublished sleight-of-hand attack to allow vote buying.

Threeballot

- We provide a mechanism to provide voters a financial incentive to vote in some desired way, when the three ballots are filled-in in a random way by some voting machine.
- We provide a technique for buying votes when the ballots are filled out manually, using a variant of chain voting.

All End to End Systems

- We report a couple of known broad categories of attack on E2E systems we did not find referenced in the literature.
- We provide a framework for vote-buying and coercion attacks.

While our attacks are specific to particular E2E systems, the general ideas behind them can be broadly applied. One goal of this paper is to get these ideas into widespread circulation, so that systems we have not considered here may also be subjected to the same analysis.

2 Background

2.1 E2E Voting Systems

Election systems in current use rely on procedures to provide integrity and ballot secrecy. Typically, voters must trust election administrators to follow these procedures. Secure elections using traditional voting systems are possible when tight controls are in place, such as maintaining the chain of custody of ballots, but it is nearly impossible for voters to gain assurance that such controls are followed. End-to-end (E2E) cryptographic voting schemes aim to provide voters with a means of verifying that elections are honest, without needing to trust election officials or that the chain of custody of ballots is maintained.

End-to-end refers to the voter's ability to verify the election from vote casting to vote counting. Most schemes operate by encoding voters' choices a special way which can be read by election officials. The encoded ballot is posted on a public bulletin board and each voter is given a voting receipt. The unique feature of E2E voting schemes is that this receipt can be used to verify the encoded ballot on the bulletin board but does not show how the voter voted.

Additionally, E2E voting schemes provide voters with a means to ensure cast ballots are counted correctly. In nearly all cases this done by having the voting scheme prove that each encrypted vote on the bulletin board is correctly decrypted, allowing anyone to verify the final vote tallies by recounting the decrypted ballots.

Front-End vs. Back-End E2E voting schemes involve a combination of activities performed by voters, election administrators and auditors. We refer to the part of the voting system that the voter interacts with the system as the *front-end*. This typically includes the ballot, receipt and bulletin board. Voters interact with the front-end to gain assurance the voting system is functioning honestly, often relying on auditors or tools to verify some parts of the voting protocol for them. The voting scheme back-end is everything that occurs partially hidden from the voter. This can include the cryptographic encoding and decodings of ballots, ballot shuffling and various third-party auditing techniques. The attacks discussed in this paper take place within the front-end of voting schemes. They involve presenting misleading information to voters that cause their votes to be miscounted, or providing voters with specific ways to interact with the front-end of the voting scheme which can be used to encourage or coerce particular votes.

2.2 Punchscan

Punchscan [6, 7] is a paper/electronic hybrid cryptographic voting scheme that uses paper ballots.[4] Each Punchscan ballot consists of two separate sheets. Voters must interact with both of these sheets to cast a vote. Viewed separately, neither sheet directly contains sufficient information to determine the selections of the voter.

The top sheet of a Punchscan ballot contains the set of ballot questions. For each question, the ballot maps each choice to a particular letter (or other symbol), along with a set of holes. Those holes line up with a permutation of the set letters which is printed on the bottom sheet. When the top sheet is stacked directly on top of the bottom sheet the voter sees each question, mappings from choices to letters, and the letter options through the holes in the top sheet.

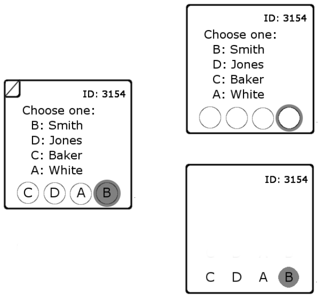


Fig. 1. A Punchscan Ballot with vote for Smith

To cast a vote, the voter looks up the letter corresponding to his or her desired response, and finds that letter by looking through the holes in the top sheet. The voter then applies a bingo dauber to that letter, thereby marking the letter on the bottom sheet and the area around the hole on the top sheet. The two sheets of the ballot are separated and the voter selects one to destroy. The remaining sheet is scanned, providing election administrators with a digital record of the vote, while physical sheet becomes the voter's receipt.[5] A representation of the digital ballot is posted on a public bulletin board, allowing the voter to compare the marks on the receipt to those appearing on the bulletin board.

Election administrators use a table containing directions for decrypting each ballot sheet to tally votes, called the Punchboard. By conducting audits before and after the election, voters can be assured that their ballots halves are correctly

[4] Recently, a related set of end-to-end voting systems have been developed under the name Scantegrity. We do not consider Scantegrity in this paper, but the mechanisms are different enough that our current attacks do not appear to apply.

[5] The current Punchscan voting procedure requires that voters select the top or bottom sheet as the receipt prior to viewing the ballot. In this paper we will propose attacks against both sets of Punchscan election procedures.

translated into their desired votes. The details of these audit procedures contain the bulk of the cryptographic techniques. However, rather than attacking the underlying cryptographic primitives of the election system, we will attack the voting procedure.

2.3 Prêt-à-voter

A detailed description of the Prêt-à-voter scheme can be found in [15]. Like Punchscan, Prêt-à-voter encodes votes based on a random permutation, in this case of the candidates. A Prêt-à-voter ballot is split between two halves, separated by a perforated edge. The left half of the ballot displays the candidates in a permuted order, while the right half has boxes that are marked to indicate a vote. Also, the right half contains a cryptographically-protected copy of the permutation of the candidates on the left side. This permutation could be encrypted using threshold cryptography or onion encryption, so only a group of election administrators would be able to decrypt that permutation. Typically, the permutation of the candidates is a cyclic shift, represented as an offset from a standard candidate ordering.

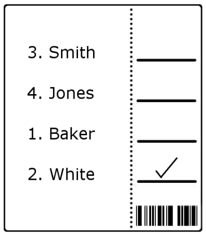


Fig. 2. A Prêt-à-voter Ballot with vote for White

To cast a ballot, voters mark the box next to the candidate of their choosing, and separate the two halves along a perforated edge. The halves containing the candidates names are destroyed, while the halves with ballot marks are scanned in and recorded as votes. Voters are allowed to bring home the right halves of ballots and compare them to the scanned copies, which are posted to a bulletin board. The cryptographic algorithms in the scheme allows voters to verify that the collection of posted ballots is properly decrypted.

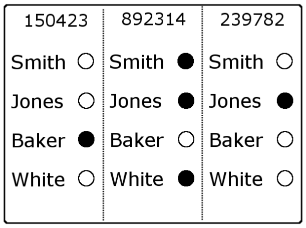
The Prêt-à-voter system relies on the proper construction of ballots. That is, the right half of the ballot must contain an encrypted representation of the permutation of candidates displayed on the left ballot half. Each voter may choose to audit a ballot by providing a special auditing device with the just the right ballot half. The device would decrypt that half and respond with the permutation of candidates. The voter could verify that this permutation matches the candidate list on the left ballot half. Auditing a ballot also invalidates that ballot, forcing a voter to obtain another ballot to either cast or audit. The attacks outlined in this paper rely on the voter being allowed to choose between casting or auditing a ballot after viewing it.

2.4 ThreeBallot

The ThreeBallot voting system [8], like Punchscan and Prêt-à-voter , uses paper-ballots. Unlike those systems, however, ThreeBallot is entirely paper based without requiring advanced cryptographic techniques to perform auditing or maintain voter privacy.

In ThreeBallot, each voter receives three identical ballots, each with a unique serial number. To vote for a particular candidate, a voter marks the candidate's name on exactly two of the three ballots. For each of the remaining candidates, the voter marks the candidate's name on exactly one of the ballots. The voter then feeds the multi-ballot into a checker which verifies that the ballot has been properly filled out. If so, the checker places a red strip across the multi-ballot and asks the voter to choose one of the three ballots to copy. The checker separates the ballots and returns them to the voter, along with a copy of the chosen ballot. The voter casts the original three ballots in a ballot box and takes the copied ballot home as a receipt.

After the election digital representations of the cast ballots are posted on a bulletin board. The voter can verify that there is a ballot on the bulletin which matches the receipt taken home. Anyone can also tally the results of the election by merely counting the votes on the ballots. The resulting tallies will be inflated by the number of voters in the election.



150423		892314		239782	
Smith	○	Smith	●	Smith	○
Jones	○	Jones	●	Jones	●
Baker	●	Baker	○	Baker	○
White	○	White	●	White	○

Fig. 3. A ThreeBallot Ballot with vote for Jones

To improve the usability of ThreeBallot, it has been suggested [8] that voters could interact with an electronic ballot marker (EBM) that would provide an interface similar to that of a DRE. After the voter selected her choices, the EBM would print out a randomly filled-in multiballot that would correspond to those choices. The voter could verify that the multiballot properly reflected her intended vote, and obtain a receipt for any one of the three ballots.

3 Election Fraud with Misprinted Ballots

The most serious threat to an election is an attack capable of changing the outcome of the election. The goal of E2E schemes is to make any changes detectable. Most E2E schemes claim to provide this, but such claims are only supported when election officials and auditors can be trusted to honestly follow proper election procedures.

The accuracy of vote counts in Punchscan and Prêt-à-voter is dependent on the proper construction of ballots. To deal with this both systems rely on pre-election audits of the ballots to ensure the ballots were created correctly. In the Punchscan scheme, election officials commit to the set of ballot forms and audit some percentage of the ballots, looking for irregularities between the actual ballot forms and the commitment on the Punchboard. Tampering with the set of ballots between the audit and the election has a good chance of being caught during the post-election audit. Here we will describe a way to tamper with the ballots in a way that would not be detected with typical audit procedures.

In this attack, a small percentage of ballots are replaced with tampered ballots. The front sheet of each of these ballots remains the same as their untampered versions, while the back sheet is changed such that the placement of two letters are swapped. Figure 4 gives an example of a tampered ballot. In this configuration, votes for Smith and Jones will be swapped. That is, a voter attempting to vote for Smith would mark the third hole, but that vote would instead be decrypted by the Punchboard to be a vote for Jones. Note that an attacker could alternatively misprint the front sheets, keeping the back sheets untampered.

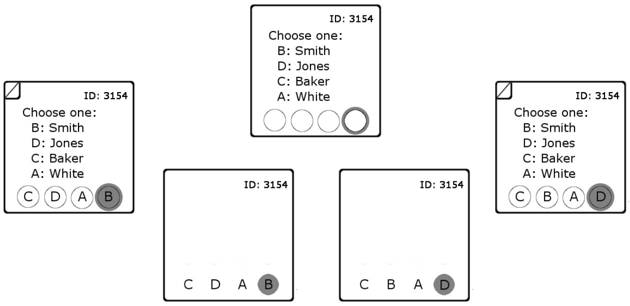


Fig. 4. Flipping Two Letters On Back Sheet

Misprinted ballot sheets are easily identifiable when compared to the Punchboard, as they will not match the committed ballots. Misprinted back sheets, when kept as a receipt, would provide evidence of election tampering. However, when the front sheet is kept as a receipt, the back sheet, the only evidence of tampering, is destroyed during the voting process. This leads to a very simple vote swapping attack when voters tell election officials what sheets they will take as receipts before the officials hand out ballots. In this case, election officials can provide misprinted ballots to only those voters who choose to keep the unmodified front sheets. This could greatly change election results if voters of a particular political party are targeted for attack.

This attack could be prevented by forcing election officials to commit to a particular ballot before asking the voter to choose a receipt sheet. However, similar attacks are still possible. In this case, attackers need to ensure that voters cannot cast a ballot and retain the modified sheet. This could be accomplished by constructing the modified back sheet such that it looks normal to the human eye but is unreadable to the scanner. Alternatively, the scanner could be involved in the attack, and recognize the ballot ID as a tampered rear ballot that should be rejected. In both cases the voter would have to obtain a new ballot and revote. This attack can be prevented by including special procedures for the handling of unscannable ballots. Unscannable ballots should be treated differently than other invalid ballots. Voters must be allowed to keep an unmarked copy of the unscannable ballot as a receipt and then be given an opportunity to vote again. During the post-election audit, election officials should open the commitment to the same half of the ballot the voter was given. The voter could use the receipt to check whether this attack occurred by comparing the opened commitment to the ballot receipt. The commitment to the other half should remain unopened in order to protect against coercion attacks.

Variations of this attack can be applied to other E2E schemes. A similar attack can be applied to Prêt-à-voter with the cooperation of ballot scanners. However, this attack appears much less likely to go unnoticed in practice. In this case, the right-hand side of the ballot would be altered by swapping the names of two candidates. Attackers would also modify ballot scanners so that they would respond with an error when modified ballots are audited.

4 Incentives, Voter Coercion and Vote Selling

Voter coercion attacks aim to give one person undue influence over another person's vote. Often, the attacker influences the voter by rewarding the voter for voting for a particular candidate or punishing the voter for failing to do so. However, as attackers can also seek to influence votes rather than force them, it is acceptable for coercion attacks to provide an incentive toward voting a particular way. This observation opens the door for a wide variety of attacks that, while not perfect, can be effective at influencing voters.

Voter coercion is inherently a contract. Voters agree to vote a particular way in exchange for a reward or to avoid punishment. A successful attack requires a way to enforce the contract. The goal of a coercion attack is to create a protocol between a coercer and voter that tends to reward voters that vote correctly. A protocol need not be perfect. It may occasionally reward voters incorrectly that voted for the wrong candidate. As long as the probability of being rewarded is higher when a voter votes correctly, the protocol will provide an incentive for voters to vote correctly. Alternatively, a protocol may occasionally fail to reward voters that vote correctly. Protocols that never fail to reward honest voters can be considered contract enforcement protocols. In this case, all voters failing to be rewarded by the protocol could be punished severely[6], knowing that only dishonest voters would be punished.

The voting system and the rules for what voters are allowed to bring into the voting booth together determine what components an attacker can use to enforce the vote buying protocol. For example, if voters are allowed to bring in cameras almost any voting system will fall to a vote buying attack. Nonetheless, it is impractical to ban everything from voting booths. A piece of paper and a pen, a pre-marked ”voters guide” or even a cell phone might be used to enforce a vote-buying contract.

4.1 Forged Ballots

A well-known attack in the end-to-end cryptographic voting community involves providing voters forged ballot halves to destroy in place of actual ballot halves. Punchscan and Prêt-à-voter ballots are split between two halves. Combined these halves display a human-readable vote, but each half on its own acts as an encrypted vote that can only be read by the election officials. The combined sheets can show anyone how the voter voted. For that reason, the election procedures of Punchscan and Prêt-à-voter require that each voter destroy one half of the ballot and retain the other half.

Voters able to leave a polling place with both halves of the ballot can use these halves to prove how they voted. A voter may be able to do this without raising suspicion from the election officials by secretly bringing a forged ballot sheet to the polls. The voter would destroy the forged ballot sheet rather than one of the original sheets. For instance, a voter may bring in a copy of the front sheet of a Punchscan ballot. After voting, the voter would slip the actual front sheet in his pocket, destroy the forged sheet in front of an election official and keep the back sheet as a receipt [7].

4.2 Incentives

Typical vote buying attacks involve an attacker paying individuals who prove that they voted for a particular candidate. However, resourceful attackers can still influence election results without learning how individuals voted by providing them with an incentive to vote a particular way. The idea is that voters will

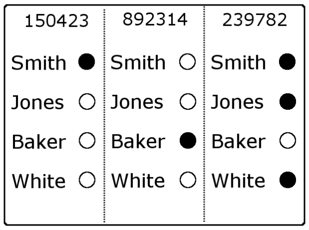
[6] A real world example of such punishment comes from the days of machine politics in the United States, where city or state employees' jobs could depend on voting the right way.

[7] Current Punchscan procedures include a clipboard lock. Each ballot is locked to a clipboard before being handed to a voter. After marking a ballot the voter tears one sheet out of the lock and destroys it, then returns to the official with the remaining sheet still locked in place. As voters, rather than election officials, destroy the ballot sheets, the clipboard locks do not prevent this attack

maximize their expected return by following the coercer's instructions. ThreeBallot, when used with an electronic ballot marker, is particularly vulnerable to incentive attacks[8]. In that variant of ThreeBallot, ballots are automatically marked by a machine. Voters make their selections on a DRE-like machine which randomly constructs a valid multiballot with votes for those selections. The attacks work on the principle that although voters cannot control the specific marks on their multiballots, their ballot choices will influence the marks.

ThreeBallot Pay-Per-Mark A simple example of an incentive attack with a machined-marked ThreeBallot device is to pay voters for each mark on a receipt that is acceptable to the vote-buyer. For example, the vote-buyer would offer to pay one dollar for every mark corresponding to a member of the Whig party. A vote-seller would cast a multiballot, choose a receipt based on the number of Whig votes contained on each ballot and present that receipt to the buyer in exchange for payment. If the seller does not vote for the Whigs on any of n total questions, each ballot would contain roughly $\frac{n}{3}$ Whig marks. However, voters who vote for Whigs on every ballot question would expect to find a ballot with $\frac{2n}{3}$ Whig marks. This is ineffective at influencing individual races and does not work when races are separated from one another.

ThreeBallot Pay-for-Receipt In many cases a vote buyer may only be interested in coercing a voter on a single ballot question. We can create an incentive in machine-marked ThreeBallot to encourage voters to vote for a particular candidate over another. Consider a close election between Smith and Jones. A vote-buyer attempting to gain votes for Smith could force voters to return with a ballot that contains a vote for Smith but not a vote for Jones. If a voter votes as directed, the voter is guaranteed to obtain a ballot that contains a vote for Smith but not Jones. However, if the voter instead casts a vote for Jones, then the voter only has a $\frac{1}{3}$ chance of obtaining such a ballot. If a neither Smith nor Jones is chosen, then the voter has a $\frac{2}{3}$ chance of obtaining such a receipt.



150423		892314		239782	
Smith	●	Smith	○	Smith	●
Jones	○	Jones	○	Jones	●
Baker	○	Baker	●	Baker	○
White	○	White	○	White	●

Fig. 5. Vote for Smith and Not Jones

[8] There are other attacks on hand-marked ThreeBallot which are not discussed here, notably the Italian attack.

Because honest voters[9] will always be able to return with the correct receipt this attack can also serve as a voter coercion attack, demanding that a voter return with the correct receipt to avoid punishment. Similar attacks can be conducted against Punchscan by extending the ideas in [20]. An attacker could develop a set of marked receipts, one of which is always obtainable if a voter votes as directed, but may not be if the voter votes for a different candidate.

Levels of Payment We can construct slightly more complicated attacks that are effective against Punchscan. Moran and Naor present a simple coercion attack against a 2-candidate race in Punchscan in [20] which allows roughly $\frac{3}{4}$ of voters to vote how they wish, but forces $\frac{1}{4}$ of voters to vote for a particular candidate. The attack works by paying for receipts marked particular ways; the ballot layout determines what the voter can do to get paid. We extend that approach here to work with multiple candidates. The basic idea is that we will pay people to vote against a particular candidate by using different levels of payouts for different receipts.

Consider a vote buyer who wants to see Smith lose an election with n candidates. The buyer would offer \$10 for any front receipt showing Smith=a with the first hole marked, or \$5 for any back receipt marked for a. If we assume voters will always act to maximize their payout, any voters receiving a ballot where Smith=a will return with the front sheet marked to randomize their vote. Thus, we know any voter returning with a marked on the back sheet did not vote for Smith. Effectively we are randomizing votes away from Smith. About $\frac{1}{n^2}$ of voters will vote for Smith, while $\frac{n+1}{n^2}$ of voters will vote for each of the remaining $n-1$ candidates, assuming voters always act to maximize their payoff.

4.3 Scratch-Off Card Attacks

Two-way communication between a voter in the voting booth and an attacker can be a very powerful tool for creating coercion attacks. In this section we will present several coercion attacks on E2E systems that work by simulating two-way communication entirely within the voting booth. This is based on similar work by Moran and Naor in [22] that used scratch-off cards to construct polling protocols. By scratching off a portion of this card based on a marked ballot, voters will permanently bind themselves to that ballot. The scratch-off card provides a challenge to the voter, which they cannot receive until after committing to a ballot form.

Basic Idea Here we will present a simplified vote selling attack where the vote buyer is in contact with a voter inside the voting booth using a cell phone. The cell phone provides a means of communicating challenges and pledges between the buyer and voter. Using the two-way communication with the vote buyer,

[9] In this context, an "honest" voter is one who votes as he's told.

and the receipt provided by the voting system, the voter is able to convince the buyer that he voted for the correct candidate.

The important observation here is that letting a voter choose one of two sheets to retain reveals as much information as letting the voter retain both sheets. This is damaging to paper-based E2E systems where a single ballot is split across multiple sheets. In the case of Punchscan, possession of one receipt and knowledge of the other (destroyed) sheet is sufficient to determine the voter's selection. This leads to the following vote-buying protocol:

1. The voter obtains a Punchscan ballot and enters booth.
2. Using the cell phone, the voter issues pledges for the two ballot sheets by telling the buyer the letters associated with each candidate (the contents of the top sheet) and the orders of the letters (the contents of the bottom sheet).
3. The buyer randomly selects one of the pledges sheets and issues this selection to the voter as a challenge.
4. The voter keeps the challenged sheet as a receipt, casts the ballot, and returns to the buyer.
5. The buyer compares the receipt to the pledged sheets from Step 2.

This protocol acts as a cut-and-choose proof of the truthfulness of the voter's pledges. Now that the buyer has obtained one sheet, and is convinced of the contents of the other sheet, it is easy to determine the vote cast.

Scratch-Off Card It might be difficult to have a cell phone conversation in the voting booth without being noticed by an election official. In this section we will discuss how to run the receipt-based vote buying protocol using a scratch-off card in place of cell phone communication. We can replace the cell communication with anything that lets a voter pledge commitments to two ballot sheets, and only then receive a challenge.

Here we will show how this can be accomplished using scratch-off cards. Suppose a group of voters agree to sell their votes to Smith. A voter can commit to the two ballot sheets by revealing the letter associated with Smith on the top ballot sheet, and the placement of that letter on the bottom sheet. We can do this on a scratch-off card with two rows of scratch-off pads. The first row will have a pad for each of the possible letters associated with Smith, the second row will have a pad for each of the possible positions. Thus, for a typical Punchscan ballot question with four candidates, our scratch off card would have a row of four pads labeled $a - d$ and a second row of pads labeled $1 - 4$.

The card needs to provide the voter with a challenge after the commitments are done. One way to do this is to have random integers under each pad. The voter would scratch off the pads associated with his top and bottom sheets, revealing two integers. The resulting sum of these integers would provide the challenge; an even sum would indicate a challenge for the top sheet, an odd sum a challenge for the bottom.

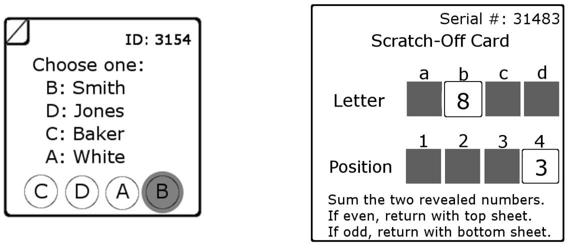


Fig. 6. A Committed Scratch-Off Card

The attack works on the principle that knowledge of the top and bottom sheets, along with placement of the mark on the ballot, is sufficient to determine the cast vote. The scratch-off card contains commitments for the top and bottom sheets, with the receipt showing the voter's mark. Voters who fill out the scratch-off cards honestly (that is, in a manner consistent with their ballots) are forced to vote for the pledged candidate otherwise their deception would be detected. A voter attempting deception would have to incorrectly fill out the scratch-off card by misrepresenting the top or bottom sheet on the card. In that case, the voter's deception would be caught if the card's challenge asks for the misrepresented sheet.

Spoiling Ballots The pledges on the scratch-off card pledge are meant to commit the voter to a particular ballot, but this commitment is weak. A determined voter might try to "cheat" the vote buying protocol by repeatedly spoiling ballots. For instance, he might vote for the wrong candidate and lie about one of the ballot sheets on the scratch off card with a 50% of being caught. If he will get caught, he could spoil ballots until obtaining one that will let him cheat. However, introducing spoiled ballots allows us to create a more flexible attack by modifying the previous protocol. In this case, voters will strongly commit to a ballot and then find out whether or not to spoil that ballot.

Spoiled ballots play an important role in many paper-based E2E systems. Punchscan and Prêt-à-voter rely on the blank ballots being properly constructed, as was discussed in Section 3. One way to give voters assurance of proper construction is to let them audit ballots on election day. This would involve posting information about the audited ballot on a bulletin board and potentially allow the voter to leave the polling place with a blank ballot. If someone were to vote on the audited ballot (which is not allowed), the auditing information posted would let anyone see how that person voted. That is what will make this an effective attack.

The basic attack is an extension of the previous attack. In the case of Punchscan, voters must still commit to both ballot sheets. They must also commit to their ballot serial numbers, in order to prevent voters from repeatedly spoiling ballots. Thus, the scratch-off card would have three rows of scratch-off pads. The first two rows would be for the letter associated with the desired candidate on the top sheet and the location of that letter on the bottom sheet. The third row would be for the last digit of the ballot serial number and would have pads for the digits $0 - 9$, perhaps with individual pads for multiple digits. As before, a random integer is underneath each pad. After scratching off the pads associated with his ballot, each voter would sum the three revealed integers. A sum congruent to 1 (mod 10) indicates the voter must spoil the current ballot and obtain a new ballot. Otherwise, the voter must cast the ballot. Either way, the voter returns to the buyer after voting and provides the scratch-off card and either the ballot receipt or the spoiled ballot. Figure 7 shows an example of a scratch-off card.

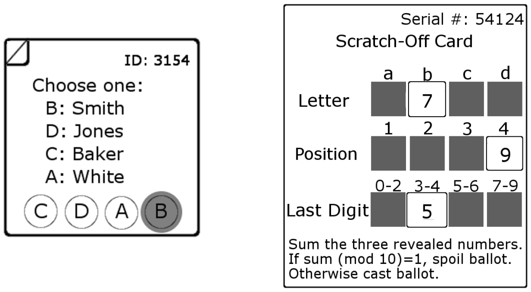


Fig. 7. A Committed Scratch-Off Card indicating spoil

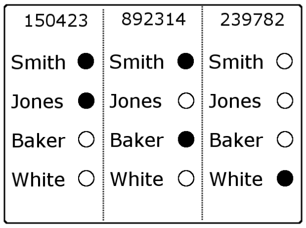
In this case, ballot spoiling serves to check that the voter filled out the scratch-off card correctly. A spoiled ballot allows the buyer to compare the scratch-off marks with the actual ballot, or some representation of that ballot on the bulletin board. Attempts at deception would be caught with 10% probability, which should be enough to voters. If the ballot is cast instead of spoiled, the buyer may assume the card was filled out correctly, and determine if the voter chose the proper candidate. For example, if the top sheet is returned, the buyer can check that the candidate-letter mapping was pledged correctly, and that the location of the mark on the receipt matches the pledged location of the candidate letter on the scratch-off card.

While this variant of the scratch-off attack, in general, detects cheating with a lower probability than the previous version, it is more flexible. Namely, this variant no longer requires the voter to choose one of two possible receipts after filling out the card. Thus, it can be used against Punchscan even when procedures force voters to choose a receipt sheet prior to viewing the ballot, a successful countermeasure against the previous attack. Furthermore, it is effective against other paper-based E2E systems, like Prêt-à-voter . In that case, voters would be given scratch-off cards that allow them to commit to the cryptographic onion on each ballot, and the placement of the desired candidate on the left-hand ballot sheet.

4.4 Beacons

The scratch-off card attacks are effective because voters must first mark their ballots a particular way then learn from a challenge whether they need to perform an action that could reveal attempts at deception. More generally, we just need a communications channel between the seller and buyer after a ballot is marked. This channel could be as simple as the buyer holding up a small sign to voters as they are about to cast their marked ballots.

Alternatively we could use a chain of coerced voters that would vote in succession. Each voter would deliver a challenge to the preceding voter. To illustrate this attack, consider a ThreeBallot election. Each coerced voter would be instructed to fill out a hand-marked multiballot as shown in Figure 8. Voters would enter the poll booths in a chain, with the buyer sending in the next voter after the preceding voter has marked their ballot. The buyer would give the voter a challenge to pass on that instructs the previous voter to return with either the left, middle or right ballot. Voters who return with the correct receipt are rewarded. Furthermore, the challenging voters are rewarded if the challenge recipients return with the correct receipt.



150423	892314	239782
Smith ●	Smith ●	Smith ○
Jones ●	Jones ○	Jones ○
Baker ○	Baker ●	Baker ○
White ○	White ○	White ●

Fig. 8. ThreeBallot Marking Instructions

5 Conclusion

Procedural changes to the voting schemes can prevent most of the attacks discussed in this paper. Many of these attacks relied on the voter being free to make a choice after viewing the ballot that would determine what information is brought back from the poll booth; e.g., which receipt to take home or whether to cast or audit a ballot. Schemes which give voters that choice are vulnerable to coercion and vote buying attacks. However, procedural defenses can create additional vulnerabilities. For instance, if election officials ask voters if they will cast or audit a ballot prior to handing them one, the official could hand out misprinted ballots to those intended to cast the ballot. This decreases the risk of a vote buying attack, but increases the risk of election fraud, a serious attack.

End-to-end voting scheme designers should be wary to rely heavily on procedures to maintain their security properties. The advantages of end-to-end voting schemes over traditional systems are reduced when they rely on procedures; optical scan systems are relatively secure when proper chain of custody is maintained with the ballots. Simple attacks, like the misprinting attack described in this paper, can target these procedures to commit election fraud or violate privacy by changing the election procedures in seemingly inconsequential ways. The chances that a procedural change will go unnoticed increases as the number of procedural controls increases. In many instances, such changes will look like simple mistakes or oversights, rather than attempts at election fraud. While it is unrealistic to imagine schemes where specific procedures need not be followed to achieve security claims, a reasonable goal is to design systems whose verifiability claims are not dependent on the actions of election administrators or third-party auditors.

The field of end-to-end cryptographic voting schemes is still relatively young. Advances in the field of cryptography such as commitment schemes, signatures, secret sharing schemes and verifiable shuffles give us a variety of tools, but there is still room to improve the protocols which use these tools and the procedures that should be followed to mitigate threats.

Acknowledgments. The authors would like to thank to Ben Adida, Bill Burr, Richard Carback, Morris Dworkin, Ben Hosp, Andy Neff, Rene Peralta, Stefan Popoveniuc, Ron Rivest, Peter Ryan, Bruce Schneier, Poorvi Vora, and David Wagner for helpful discussions and comments on the topic of this paper.

References

1. Norden, L.: The machinery of democracy: Protecting elections in an electronic world. Technical report, Brennan Center Task Force on Voting System Security (October 2006) `http://www.brennancenter.org/dynamic/subpages/download_file_38150.pdf`.
2. NIST: Threats to voting systems workshop (2005) `http://vote.nist.gov/threats/`.
3. California Secretary of State: Top-to-bottom review (2007) `http://www.sos.ca.gov/elections/elections/elections_vs.htm`.
4. Neff, C.A.: Practical high certainty intent verification for encrypted votes. VoteHere (2004) `http://www.votehere.net/vhti/documentation`.
5. Adida, B., Neff, C.A.: Ballot casting assurance. In: EVT'06: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop, Berkeley, CA, USA, USENIX Association (2006) 7–7
6. Popoveniuc, S., Hosp, B.: An introduction to punchscan. In: Proceedings of Workshop on Trustworthy Elections (WOTE),. (2006)
7. Fisher, K., Carback, R., Sherman, A.: Punchscan: Introduction and system definition of a high-integrity election system. In: Proceedings of Workshop on Trustworthy Elections (WOTE),. (2006)
8. Rivest, R.: The threeballot voting system. MIT (2006) `http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf`.

9. Adida, B., Rivest, R.L.: Scratch & vote: self-contained paper-based cryptographic voting. In: WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society, New York, NY, USA, ACM Press (2006) 29–40
10. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2) (1981) 84–90
11. Cohen, J.D., Fischer, M.J.: A robust and verifiable cryptographically secure election scheme. In: Proc. 26th IEEE Symp. on Foundations of Comp. Science, Portland, IEEE (1985) 372–382
12. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: ASIACRYPT '92: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, London, UK, Springer-Verlag (1993) 244–251
13. The Center for Responsive Politics: Opensecrets `http://www.opensecrets.org`.
14. Kelsey, J.: Strategies for software attacks on voting machines. Threats to Voting Systems Workshop (2005) `http://vote.nist.gov/threats/papers/stategies_for_software_attacks.pdf`.
15. Chaum, D., Ryan, P.Y.A., Schneider, S.A.: A practical voter-verifiable election scheme. In di Vimercati, S.D.C., Syverson, P.F., Gollmann, D., eds.: ESORICS. Volume 3679 of Lecture Notes in Computer Science., Springer (2005) 118–139
16. Rivest, R., Smith, W.: Three voting protocols: Threeballot, vav and twin. In: Proceedings of USENIX/ACCURATE Electronic Voting Technology Workshop (EVT). (2007)
17. Strauss, C.: The trouble with triples: A critical review of the triple ballot (3ballot) scheme. part 1. Verified Voting New Mexico (2006) `http://www.cs.princeton.edu/~appel/voting/Strauss-TroubleWithTriples.pdf`.
18. Strauss, C.: A critical review of the triple ballot voting system. part 2: Cracking the triple ballot encryption. draft version 1.5. Verified Voting New Mexico (2006) `http://www.cs.princeton.edu/~appel/voting/Strauss-ThreeBallotCritique2v1.5.pdf`.
19. Clark, J., Essex, A., Adams, C.: On the security of ballot receipts in e2e voting systems. In: Proceedings of Workshop on Trustworthy Elections (WOTE),. (2007)
20. Moran, T., Naor, M.: Split-ballot voting: Everlasting privacy with distributed trust. In: Proceedings of Workshop on Trustworthy Elections (WOTE),. (2007)
21. Essex, A., Clark, J., Carback, R., Popoveniuc, S.: The punchscan voting system: Vocomp competition submission (2007) `http://www.punchscan.org/vocomp/PunchscanVocompSubmission.pdf`.
22. Moran, T., Naor, M.: Polling with physical envelopes: A rigorous analysis of a human-centric protocol. In (Editor), S.V., ed.: EUROCRYPT 2006. Volume 4004 of Lecture Notes in Computer Science., Springer-Verlag (May 2006) 88–108