CALTECH/MIT
VOTING TECHNOLOGY PROJECT

A multi-disciplinary, collaborative project of
the California Institute of Technology – Pasadena, California 91125 and
the Massachusetts Institute of Technology – Cambridge, Massachusetts 02139

THE THREEBALLOT VOTING SYSTEM

Ronald L. Rivest
MIT

Key words: ***paper-based voting, voting security, paper ballots, vote verifications***

VTP WORKING PAPER #56
October 2006

The ThreeBallot Voting System

Ronald L. Rivest
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
rivest@mit.edu

October 1, 2006*

Abstract

We present a new paper-based voting method with interesting security properties. The attempt here is to see if one can achieve the same security properties of recently proposed cryptographic voting protocols, but without using any cryptography, using only paper ballots. We partially succeed. (Initially, I thought the proposal accomplished this goal, but several readers discovered a vote-buying attack (see Section 4.4) that appears to be rather difficult to fix without making the resulting system much less usable in practice. Currently, this paper should thus be viewed more as an academic proposal than a practical proposal. Perhaps some variation on these ideas in this paper might still turn out to be of practical use. The "OneBallot with Exchanged Receipts" system sketched at the end of Section 5.3.1, looks particularly promising at the moment...)

The principles of ThreeBallot are simple and easy to understand.

In this proposal, not only can each voter verify that her vote is recorded as she intended, but she gets a "receipt" that she can take home that can be used later to verify that her vote is actually included in the final tally. Her receipt, however, does not allow her to prove to anyone else how she voted.

In this "ThreeBallot" voting system, each voter casts *three* paper ballots, with certain restrictions on how they may be filled out, so the tallying works. These paper ballots are of course "voter-verifiable." All ballots cast are scanned and published on a web site, so anyone may correctly compute the election result.

A voter receives a copy of *one* of her ballots as her "receipt", which she may take home. Only the voter knows which ballot she copied for her receipt. The voter is unable to use her receipt to prove how she voted or to sell her vote, as the receipt doesn't reveal how she voted.

A voter can check that the web site contains a ballot matching her receipt. Deletion or modification of ballots is thus detectable; so the integrity of the election is verifiable.

*The latest version of this paper can always be found at http://theory.csail.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf

1 Introduction

Designing secure voting systems is tough, since the constraints are apparently contradictory. In particular, the requirement for voter privacy (no one should know how Alice voted, even if Alice wants them to know) seems to contradict verifiability (how can Alice verify that her vote was counted as she intended?).

The proposal presented here is an attempt to satisfy these constraints *without* the use of cryptograpy. We get pretty close...

Like most cryptographic proposals, ThreeBallot uses a public "bulletin board"–a public web site where election officials post copies of all of the cast ballots (there will be $3n$ of them if there are n voters) and a list of the names of the voters who voted. (Some states might use voter ID's rather than voter names.)

One key principle of ThreeBallot is to "vote by rows" and "cast by columns". The ThreeBallot ballot can viewed as an array, where the voter places marks in rows corresponding to candidates, but then separates the columns and casts them separately, keeping a copy of one.

ThreeBallot provides a nice level of end-to-end verifiability—the voter gets assurance that her vote was cast as intended and counted as cast, and that election officials haven't tampered with the collection of ballots counted.

2 Background

We assume that the reader is somewhat familiar with voting systems. For more background, the following readings are recommended:

- Roy Saltman's new book, *The History and Politics*

of Voting Technology [19] is an outstanding scholarly history of the evolution of voting technology.

- Andrew Gumbel's book *Steal This Vote* [9] is an excellent, entertaining, and very readable review of election fraud in America.
- The Brennan Center for Justice has published an excellent report [1] on voting system security, with detailed discussions of specific threats and assessments of the risks they represent.
- Randell and Ryan's recent excellent article, "Voting Technologies and Trust," [15], which, like this paper, explores paper-based voting system architectures similar to those of cryptographic voting systems.
- Ben Adida's recent PhD thesis [3] (particularly Chapter 1) reviews voting system requirements and cryptographic voting systems, before giving improved cryptographic voting systems.
- There are numerous web sites with information and links about voting and voting technology, such those of Doug Jones [10], myself [16], the CalTechMIT Voting Technology Project [14], ACCURATE [2], or the Election Assistance Commission [7], to name just a few. (Try googling "voting technology".)

3 Details

We now describe the ThreeBallot voting system in more detail.

3.1 Checking In to the Poll Site

Each voter identifies herself as usual at the poll site, and then gets a paper "multi-ballot" to vote with. (For convenience in this exposition, the voter will always be feminine.)

3.2 The Multi-Ballot

The multi-ballot consists of three ballots. They may be ballots on separate sheets of paper, or three ballots printed on a single sheet of paper, with perforations to allow later separation. For ease of exposition, we assume that later arrangement for now.

ThreeBallot is perhaps most easily viewed as a variation or extension of "mark-sense" ("optical scan" or "opscan") systems [11].

In this arrangement, the multi-ballot has three columns, each of which is a complete ballot. Each ballot is identical, except that the ballot ID number on the bottom of each ballot is unique. There are vertical perforations between the ballots, so they can be separated. See Figure 1.

Each ballot has two parts: the upper "voting region," and then the "ballot ID region" on the lower part.

The voting region of a ballot contains the candidate names, each with an op-scan bubble that can be filled in by the voter.

Each ballot has a distinct ballot ID, different from the ID's of other ballots on its multi-ballot and from all other ballot ID's. The ballot ID's on the three ballots of a multi-ballot are unrelated in any way to each other, they are merely randomly assigned unique ballot ID's, with no cryptographic or other significance. The ballot ID might be a long (e.g. 7-digit) number which is essentially random, or some other unique identifier, possibly in barcoded form. For now, we'll assume that the ballot ID's are pre-printed on the ballots, but we'll see that there are security advantages to having them added later instead by the voter or by the "checker" (see Section 3.4).

3.3 Filling Out The Multi-Ballot

The voter is given the following instructions for filling out the multi-ballot. See Figure 2 for an example of a filled-out multi-ballot.

- You have here three optical scan ballots arranged as three columns; you will be casting all three ballots.
- Proceed row by row through the multi-ballot. Each row corresponds to one candidate. There are three "bubbles" in a row, one on each ballot.
- To vote **FOR** a candidate, you must fill in exactly two of the bubbles on that candidate's row. You may choose arbitrarily which two bubbles in that row to fill in. (It doesn't matter, as all three ballots will be cast.)
- To vote **AGAINST** a candidate (i.e., to not vote **FOR** the candidate, or to cast a "null" vote for that candidate), you must fill in exactly one of the bubbles on that candidate's row. You may choose arbitrarily which bubble in that row to fill in. (It doesn't matter, as all three ballots will be cast.)
- You *must* fill in *at least one* bubble in each row; your multi-ballot will not be accepted if a row is left entirely blank.
- You *may not* fill in *all three* bubbles in a row; your multi-ballot will not be accepted if a row has all three bubbles filled in.
- You may vote **FOR** at most one candidate per race, unless indicated otherwise (In some races, you are allowed to vote **FOR** several candidates, up to a specified maximum number.) It is OK to vote **AGAINST** all candidates.

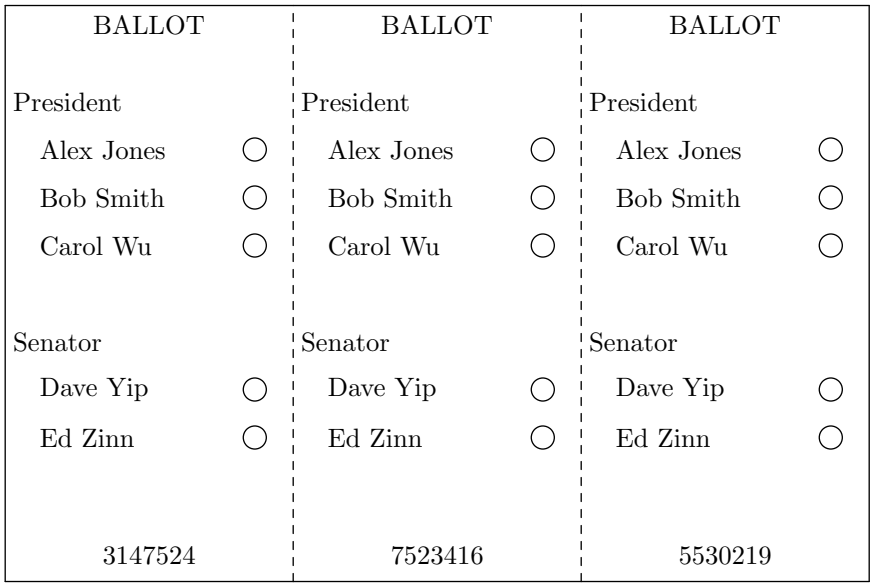

Figure 1: A sample ThreeBallot multi-ballot, with a first race for President with candidates Jones, Smith, and Wu and a second race for Senator with candidates Yip and Zinn.

- The above rules are complete. Don't worry about how an individual ballot looks—it may have a number of filled-in bubbles for a given race that is less than, or greater than, one.

We will call a filled-in bubble a *mark* (for the candidate of that row).

Note that any one of the three ballots, when viewed in the conventional way, might be an "overvote" (more than one candidate marked in a race), or an "undervote" (no candidates marked in a race). That is OK here. But when the three ballots are viewed together, they must be properly filled out, with at most one candidate per race having two marks and all other candidates having exactly one mark.

3.4 Checking the Filled-Out Multi-Ballot

When a voter has so indicated her choices, she inserts her multi-ballot in the "checker machine," whose primary purpose is to check the validity of her multi-ballot.

The checker machine might be in the voting booth, or somewhere in the middle of the voting area.

The checker checks that the voter has made exactly one or two marks for each candidate (the "row constraints"), and has made two marks for at most one candidate in each race (the "race constraints").

Note that for the race constraints to be checked, the checker needs to have a description of the structure of the ballot style—where races start and end on the ballot, and how many candidates can be voted for in a given race.

If the multi-ballot is invalid, the machine beeps and indicates where the voter has put too few or too many marks.

If the multi-ballot is OK, the machine beeps (now in a nice way) and puts a horizontal red stripe across the bottom of the multi-ballot (below the ID's). (It may also print other information on the ballots, such as authenticating information.)

When the checker spits back a correct multi-ballot, it also cuts it into three separate ballots along the perforations.

Once the red stripe is there, the multi-ballot must then be cast, as three separated ballots. (This is enforced by procedures at the poll site.)

The checker machine makes no recordings of what it has seen; it is stateless and pretty "dumb" (although the checker does need to be able to accept as input a description of the ballot style, as noted above, in order to check the race constraints).

(A "really dumb" checker would be election ballot-style independent, and thus would check only the row constraints. Such a checker might have some utility... It is also interesting to consider somewhat more complex checkers, such as the "Shamos checker" of Section 5.2.2, which prints ballot ID's on the ballots, etc.)

The voter now has three separated ballots.

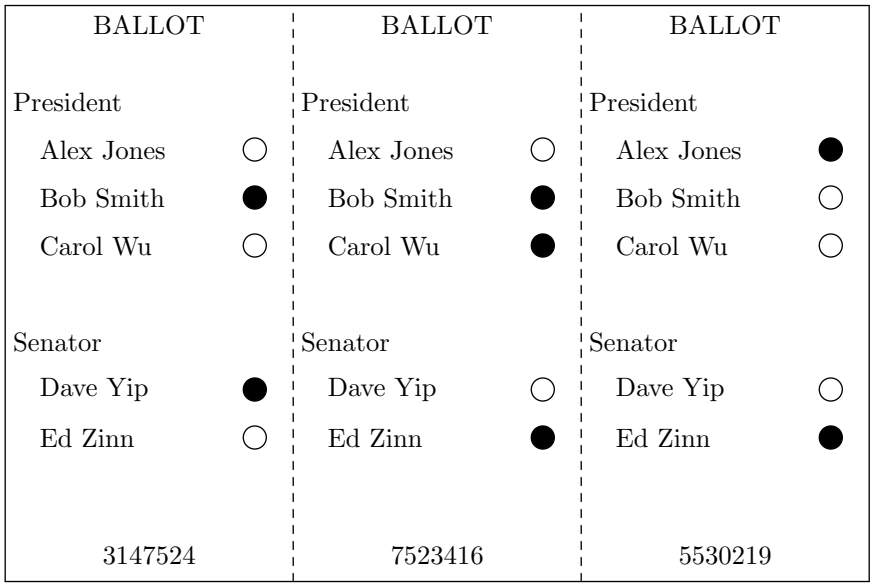

Figure 2: A filled-out version the multi-ballot of Figure 1, showing a vote **FOR** Smith for President and a vote **FOR** Zinn as Senator, since the rows for these candidates have two filled-in bubbles (marks) each. All other rows have exactly one mark. (There are many other ways such choices could have been indicated.) Note that ballot 7523416, when viewed as a conventional ballot, looks like an overvote for President.

3.5 Getting a Ballot Copy as a "Receipt"

The voter should also receive a copy of one of her three ballots as her take-home "receipt". It is important that she is allowed to choose secretly and arbitrarily which of the three ballots she receives a copy of. The receipt could be printed on yellow paper, say, so it looks different than her three original ballots.

Rather than having a separate copying station, which has certain security risks, the copier is best embedded in the checker. That is, the checker should also be a (partial) copier—if the multi-ballot checks out OK, then the voter pushes button "1", "2", or "3" on the checker to get her three (now separated) ballots and also the copy of her indicated ballot. Which of the three ballots she choses to get a copy of should be known only to her.

The voter should of course check that her receipt indeed matches its corresponding ballot.

3.6 Casting Three Ballots

The voter now casts all three original ballots by putting them separately into the ballot box. The ballot box has the property, as usual, that it effectively scrambles the ballot order, destroying any indication of which triple of ballots originally went together, and what order ballots were cast in.

Note how ThreeBallot follows the philosophy of "vote by rows, cast by columns"—each candidate's vote is in a given row, but the ballots are columns. Each ballot by itself (and thus the receipt that the voter takes home) contains no information about the whether the multi-ballot was a vote **FOR** or **AGAINST** the candidate.

3.7 Going Home

The voter takes her receipt, and goes home. (In one interesting variation of ThreeBallot, discussed in Section 5.3.1 below, the voter may, on her way out of the polling site, exchange her receipt for that of another voter's.)

3.8 Posting the Ballots

At the end of election day, the entire set of ballots cast is scanned and published on the bulletin board.

(The scanning process produces only a "compact" representation of the voter's choices that just records the marks present and the ballot ID. A pixel-level scan is not used because it would introduce the problem that a voter could put stray scribbles in the margin of the ballot that would allow her to identify her ballot image later.)

The election officials will also post on the bulletin board a list of the names of all voters who voted in the election.

3.9 Checking the Integrity of the Bulletin Board

A voter can check later, after she has gone home, if she wishes, that her receipt does indeed match a ballot posted on the bulletin board. (She could also give her receipt, or a copy of it, to someone else, to check for her, if she doesn't have a convenient way to access the bulletin board, or is too busy to do so herself. See Adida [3] for a discussion of helper organizations.)

If the bulletin board doesn't contain a ballot matching her receipt, she can take her receipt to an election official and file a protest. There should be a time limit (e.g. two days) on when such protests can be filed. The election official may examine the voter's receipt to determine its authenticity, and may authorize a rescan of the cast paper ballots.

3.10 Tallying and Announcing the Winners

The ballots can be tallied by anyone, since they are publicly posted in "plaintext" on the bulletin board. (No decryption is needed, as is the case with cryptographic voting systems.)

The winners can be announced.

Note that each voter has effectively marked exactly once or twice for each candidate (due to the "row constraints"), and has marked twice for at most one candidate in each race (due to the "race constraints"); the checker has enforced these properties.

So each candidate's tally will be "as usual," except that each total is inflated by the number of voters. The election outcome is the same.

For example, if there are n voters, and candidates A, B, and C would ordinarily have received a, b, and c votes, respectively, then with ThreeBallot the final tallies will show $n + a$, $n + b$, and $n + c$ marks for A, B, and C, respectively; the vote totals for each candidate can be obtained by subtracting the number n of voters from the total number of marks for that candidate.

This completes our description of the operation of the ThreeBallot voting system. We now turn to a discussion of its security.

4 Security — Integrity

Of the two main voting system security requirements—(1) integrity of election results and (2) voter privacy—we begin with the first, since it is arguably more important.

The voter can check that

- a ballot matching her receipt is posted on the bulletin board in the list of cast ballots, and
- the total number of ballots on the bulletin board is three times the number of voters who voted (the list of voters who actually voted is also published on the bulletin board).

These checks (particularly the first), don't have analogues in most current voting systems. They allow detection of several kinds of fraud, as we shall see. Of course, one has to be careful, when one adds new security mechanisms, that they can't themselves be easily attacked. So we also consider here attacks on these additional checks. Nonetheless, one shouldn't lose track of the fact that these new checks in any case will be providing an increased level of integrity and security, compared to systems where there are no comparable checks. That is to say, these new checks are effectively another layer of defense against attacks on the voting system; other currently-used security mechanisms (e.g. for opscan systems) aren't replaced, merely augmented. Security can only get better.

4.1 Adding Ballots can be Detected

An adversary can't increase the number of ballots on the bulletin board without simultaneously putting more voter names on the bulletin board, which should be detected by someone, somehow (Grandma, did you really vote? Weren't you sick that day?)

4.2 Modifying or Deleting Ballots can be Detected

An adversary can't delete or modify any posted ballots, without risking a voter protesting that her receipt isn't matched by a ballot on the bulletin board. (Either it isn't there, or a ballot with the right ballot ID number is there, but its marks are missing or different.)

Of course, an adversary might risk modifying just a few ballots, hoping to avoid detection. But any large-scale fraud would get detected, with even a low level of vigilance on the part of voters or their proxies.

Since attacks by adding, modifying, or deleting ballots are detectable, voters can have confidence in the correctness of the finaly tally.

4.3 No Voter Coercion or Vote Selling

One design goal of the ThreeBallot system is that the voter should not be able to sell her vote, since her receipt doesn't bear any reliable information on how she voted.

Note that no matter whom she votes for, her receipt can have any possible pattern of marks. (There are 2^r such marking patterns for the receipt if there are r rows on a ballot, from completely empty to completely filled.)

Moreover, the voter has complete control over what pattern of marks are shown on her receipt.

A coercer can pay the voter to come back with a receipt showing some particular pattern of marks, and the voter can do so, *without affecting her ability to vote in any way she chooses*. She can put the coercer's desired pattern of marks in ballot 1, and then fill in ballots 2 and 3 to achieve her desired voting pattern. Ballots 2 and 3 can "outvote" ballot 1 as necessary. She then takes a copy of ballot 1 away as her receipt to give to the coercer for cash.

4.4 The "Three-Pattern" Attack

But there is another attack that might allow a coercer to buy votes or influence a voter's voting behavior.

[This is the attack that I overlooked in earlier versions of this paper, and which was pointed out to me by several readers (thanks!).]

In this "three-pattern" attack, the adversary pays the voter to vote according to *pre-specified patterns in each of her three ballots*. That is, the adversary isn't paying for the voter's "net vote", but paying for her to create her net vote in a *specific pattern of three individual ballots*. If the adversary doesn't see *all three* pre-specified ballots posted on the public bulletin board, the voter doesn't get paid (or perhaps the voter is punished somehow by the adversary).

This attack is particularly troublesome, and in the end, it makes ThreeBallot much less attractive than I had originally hoped for.

There are nonetheless several ways one can try to prevent this attack or mitigate its effects:

- The multi-ballots may come pre-printed with one mark already randomly placed in every row.

- The voter may use a DRE/EBP (DRE/electronic ballot printer) to create and print her multi-ballot see Section 9.3. The voter enters her choices on a touch-screen. The DRE controls the random allocation of marks in each row.

- A voter can conspire with two other voters to ensure that a particular triple of ballots appears on the bulletin board, without constraining how any of the three voters votes.

- Use the "cell-based approach" described in Section 5.3.1, wherein each ballot now corresponds to an individual cell of the $r \times 3$ array, rather than corresponding to a complete column of that array.

Each of these approaches itself has problems. From a theoretical point of view, the cell-based approach is probably the best. From a practical point of view, using a DRE/EBP may be best.

4.5 Recounts and Audits

Because the ballots are cast in paper form, it is possible to rescan and recount them in that form. A recount of some precincts might be mandated by state law, particularly for close elections. Or, a recount might be triggered if sufficiently many voters (or any voters!) credibly claim that their receipts aren't represented correctly or at all on the bulletin board.

4.6 Detecting Malicious Voters

The receipt may need some additional authentication, as usual here (e.g. Adida [3, Section 5.3]), to prevent voters from maliciously claiming that their (fabricated) receipt doesn't match any ballot on the bulletin board. This authentication could take the form of a seal or sticker on the receipt, perhaps on the back, or a digital signature on the receipt.

Note that it is OK for the voter to show her receipt to an election official, so that the official can mark it as an officially approved receipt, since voter privacy isn't threatened. However, the election official should *not* record the ID number of her receipt.

4.7 Attacking the Checker

The checkers need to be tested carefully. A maliciously modified checker might allow voters voting for a particular candidate to cast three marks for a candidate, and/or no marks for another candidate. Voters who can cast ballots violating these "row constraints" would then have more "weight" than ordinary voters.

An ordinary voter can increase the total number of marks cast for her favored candidate by one mark, relative to the total number of marks cast for other candidates, since she casts two marks for her favored candidate and one mark for other candidates.

If the checker is corrupt or defective, a voter may be able to increase the relative number of marks cast for her favored candidate by *three* marks, since she could be allowed to cast three marks for her favored candidate and none for the other candidates. This is obviously in violation of a principle of democratic voting—each voter should have an equal influence on the result.

Note that such illegal voting patterns can't generally be detected later or during a recount, since the "row constraint" can't be tested again once the multi-ballot is split into separate ballots.

(Of course, in some cases you may be able to tell that such an attack has been mounted. If a candidate ends up with a mark total of more than $2n$, or less than n, where n is the number of voters, then some row conditions must have been violated. But often you just won't be able to tell, once the multi-ballots have been separated.)

Thus, we see that there is some dependency of the correctness of the election outcome on the correctness of the checker (assuming that some voters would exploit an opportunity afforded by a defective checker).

On the other hand, the row conditions are exceptionally simple to check, and a simple hard-wired mechanism for checking them in the checker may be sufficiently trustable that one can have confidence in their correct operation on election day.

It is interesting to compare this situation with other forms of VVPR (voter-verified paper records), such as DRE-VVPAT or ordinary opscan ballots. ThreeBallot is like these other forms of VVPR, in that the voter can directly verify her own paper ballot (or multi-ballot), to ensure that her own intent has been captured correctly on paper. But with ThreeBallot there is an additional risk, that if the checker has been corrupted, then *other* voters might be able to cast a "heavier" vote than hers. The corresponding risks with the more traditional forms of VVPR are the usual forms of "casting multiple votes," and "ballot stuffing", wherein someone may add many extra votes (not just two extra votes) to the ballot box. Also, ThreeBallot allows voters to detect modification of the collection of cast ballots, whereas the other VVPR schemes don't even try to detect such attacks. I think on balance ThreeBallot addresses better the more serious threats.

It is also interesting to compare this situation with cryptographic voting schemes. A bad checker in ThreeBallot might allow a voter to cast an invalid multi-ballot; cryptographic schemes either make such invalid voting impossible or require the voter to post with her vote a proof that her (encrypted) ballot is valid.

We also note (as pointed out by John Kelsey) that a maliciously modified checker, since it knows which ballot is being copied for the receipt, might be able to encode this information on the ballots themselves (say by using a bit of steganography in the way the red stripe is placed on the ballots); a correspondingly corrupted scanner would then know which ballots it could scan incorrectly. This sort of michievous behavior also needs to be prevented, by design of the checker, or by other controls.

4.8 Paying for Receipts

Another potential vulnerability occurs if the adversary is able to buy (or otherwise obtain) the receipts of voters as they leave the poll site, and if the adversary also has the ability to manipulate the contents of the bulletin board. In general, there is a problem if an adversary can modify the bulletin board *and* can find out somehow what the serial numbers are on the receipts.

In this attack the adversary pays the voter to surrender her receipt, as she leaves the polling site.

Then, knowing that the voter has now given up her ability to contest the corruption of the corresponding posted electronic version of this ballot, the adversary can modify the posting on the bulletin board of her corresponding ballot.

This attack is somewhat complex and difficult to mount, but not impossible.

Some techniques are available for reducing the risk of such an attack.

Voters should be cautioned not to casually discard or give away their receipts. (The adversary would be happy to take for free via "dumpster diving" what he might otherwise pay for.) If the voter uses a "helper organization" (e.g. the League of Women Voters) as a proxy to check bulletin board integrity, the voter might deposit with the helper organization just a copy of her receipt, rather than the original receipt. She can even deposit copies with several helper organizations.

There should be strong safeguards on bulletin board modification; this provides a "layered defense". Indeed, current voting systems rely for their security *entirely* on such safeguards, since they provide voters with no capability whatsoever to verify the contents of the official list of cast ballots. In this sense, ThreeBallot can't help but be an improvement over current systems. Even if an adversary can buy or obtain receipts, that only puts him in a position comparable to what he would be in with current voting systems.

Another approach for defeating this attack is for voters to retain an extra secret copy of their receipt—the adversary thinks he is getting the only copy of the receipt, but in fact he is not. If the receipt is signed at the poll site by an election official with a digital signature represented in scannable form (e.g. a bar-code), then a copy of the receipt is as good as the original as far as protesting goes, so the adversary can't "take away" the voter's ability to protest if she retains a secret copy of her receipt.

This attack also works for many of the cryptographic schemes in the literature; the only prior treatments of this attack to my knowledge are by Ryan and Peacock [17, Section 5.4], who suggest both voter education and having election officials keeping additional copies of the receipts at the polling site, and by Karlof et al. [13, Section 5.2], who suggest voter education.

Other, more exotic approaches (such as "range voting"—see Section 9.7) could also mitigate the potential damage an adversary could do through this sort of attack, since the vote tallies would then be on the average much larger, and changing only one ballot per vote would have a smaller relative effect.

4.9 Chain Voting

Any paper-based voting system needs to consider the possibility of a "chain-voting" attack (see Jones' excellent description [12]). The usual remedies–e.g. mech-

anisms to ensure that voters actually cast the ballots they were originally given–are applicable here as well. See Jones [12] for details.

5 Security — Voter Privacy

We now turn to the second main security requirement for voting: maintaining voter privacy.

The first of Professor Michael Shamos's "Commandments" [20] on voting is:

> *Thou shalt keep each voter's choices an inviolable secret.*

Even if the voter wishes to violate her own privacy, there should be no way for her to do so. She should not be able to convince anyone else that she voted in a particular way—otherwise she could sell her vote. (This is one reason why I strongly favor pollsite voting, with its enforced voter isolation during voting, over remote voting schemes such vote-by-mail, vote-by-phone, or vote-by-Internet.)

What evidence could the voter give to an adversary, in an effort to convince the adversary of how she voted? The are three sorts of evidence available:

- physical evidence the voter brings away from the the voting session (such as her voting receipt),
- other evidence the voter may bring away from the voting session (such as ballot ID's she may have memorized or photographed), and
- the bulletin board of all cast ballots.

5.1 Does the Receipt violate Voter Privacy?

We note that there is nothing to prevent a voter from, when she wishes to vote "**FOR**" a candidate, always marking just the first two ballots, and, when she wishes to vote "**AGAINST**" a candidate, marking just the third ballot. If she takes a copy of her first (or second) ballot as her receipt, the marks on her receipt indicate exactly how her votes will be tallied.

But her receipt is at best only a "reminder" of how she voted, not a proof that will convince anyone else as to how she voted. The voter is unable to intentionally violate her own privacy by showing someone else her receipt. (We have already argued, in Section 4.3, that a voter's receipt, by itself, bears no information about how a voter voted. So, the receipt, by itself, does not violate voter privacy.)

5.2 Can the Receipt be linked with its other two ballots?

There should be no way for anyone to be able to reliably and convincingly link together the three ballots on the bulletin board that together constitute an original cast multi-ballot. If this could be done, then the voter's privacy is at risk, since the ballot ID on her receipt can allow identification as to which linked triple of ballots (a reconstituted multi-ballot) is hers, revealing her vote.

5.2.1 Risk at the Printers

We should ensure that no-one knows ahead of time what triples of ID's constitute a multi-ballot.

There is potentially a risk to voter privacy at the printing establishment, if someone there records which ballot IDs were printed on the same multi-ballot. (This problem is shared with some cryptographic voting schemes, such as Prêt à Voter.)

Some approaches to mitigating or eliminating this threat are:

- Procedural controls at the printers and within ballot delivery, to ensure that inappropriate records are not kept.
- Not printing combined multi-ballots, but only individual ballots, each with their own ballot ID as usual. Then the voter assembles a multi-ballot by randomly picking three ballots from the collection of blank ballots when she signs in to vote.
- Having the voter add the ballot ID's to the ballots using individual ballot ID "stickers" drawn randomly from a bin of such stickers. (All the ID numbers within a bin being distinct, of course.) The stickers could have the ID numbers under scratch-off, if you like, although the scratch-off needs to be removed from one ballot just before it is copied to make a receipt, and from the other two as well just after the ballots are cast but just before they are scanned for posting on the bulletin board.
- Having the checker add the ballot ID's to the ballots (see the description of the "Shamos checker" in the next section).

5.2.2 Risk That the Voter can identify her Multi-Ballot

The voter should not be able to record or remember the ballot ID numbers of her three ballots, at least not in a believable manner. (Voters should not be allowed to take photos of their multi-ballot with a camera or cellphone!) Some approaches towards achieving this goal include:

- Printing the ballot ID's in a 1D or 2D bar-code, which is hard for the voter to parse and remember.

Of course, this makes it hard for the voter to read the ballot ID later when she wants to look it up on the bulletin board.

- Printing the ballot as a long string of digits or letters, most of which are fixed "noise" and only a few encode the varying ballot ID. For example, a ballot ID might look like:

  8524710045631106558731 45

  a string of length 25, where only digits in positions 4,7,11, 15, 19, and 22 vary and and used to represent the ballot ID. (Can you remember enough about the above ID to recreate it given the following ID?)

  8524711045831176559761 45

- The ballot ID's could be under "scratch-off," as noted above.

- (The "Shamos checker".) Michael Shamos suggested the following nifty approach, which prevents the voter from ever seeing the ballot IDs of the two ballots not copied to make a receipt:

  - All multi-ballots are initially identical and contain no ballot ID's.
  - When the checker determines that a multi-ballot is valid, it prints three randomly generated ballot ID's on the three ballots, but retains the ballots for now.
  - The voter selects which ballot she wishes to have copied for her receipt.
  - The checker spits out both the selected ballot and a copy of it (her receipt), and puts the other two ballots into a holding bin.
  - She checks that the receipt and the selected ballot are identical. If so, she puts the selected ballot into the ballot box and presses the "Done" button on the checker, which empties the holding bin (containing her other two ballots) into the ballot box, in such a way that she never sees their ballot ID's. If not, she pushes the "I got a bad receipt" button on the checker (which now empties the holding bin with her other two ballots into a spoiled ballot bucket), complains to a pollworker by showing the selected ballot and unequal printed receipt, and votes again.

  The "Shamos checker" keeps the voter from ever seeing the ballot ID's of her other two ballots, so we don't need to worry about her memorizing them or photographing them! It is also consistent with state laws (like California's) that require all blank ballots to be identical.

5.2.3 Risk of Copying

It is probably better not to give the voter access to a generic copying machine in order to make a copy of one of her three ballots. This would risk that the voter makes a copy of all three of her ballots.

There should be procedural or mechanical controls to ensure that the voter only gets a copy of one of her three ballots, and can't use the copier to copy her other ballots or their ballot ID's.

The approach described earlier, where the copier is embedded in the checker machine, is perhaps the simplest way to enforce such copying limits. The checker machine should refuse to produce a copy of a ballot if it has already been checked (e.g. if it contains the red "checked OK" stripe).

Another approach (suggested by Silvio Micali), might be to have the ballots come automatically attached with carbonless copying paper underneath, so copies are made automatically while the voter votes. No copying machine is needed then, but you need to ensure that the voter only takes away only one of the three copies made.

5.3 Risk of ballot modification before casting

There is also a risk that a voter might modify her ballots, after they have been approved by the checker, but before they are cast into the ballot box.

If she makes additional marks on her ballots, or erases marks on her ballots, after they have been approved by the checker but before she casts them into the ballot box, she may be able to commit election fraud without detection, since the row and race constraints are no longer checkable once the ballots are split up.

The natural mechanisms for defeating such attacks include:

- making sure the voter doesn't handle the ballots after the checker approves them (some conveyance would be needed to get the ballots to the ballot box),

- including along with the red stripe some "checksum" information on the ballot (possibly a barcoded representation of the marks that are supposed to be on the ballot) that would be difficult for the voter to manipulate.

5.3.1 The Reconstruction Attack

In a "reconstruction" attack the adversary examines all possible triples of ballots from the bulletin board, and determines which of them form legal ThreeBallot ballots.

The information gained may, in some cases, be sufficient to determine how an individual voter voted, when

taked together with the ballot ID available on the voter's receipt.

The problem here is that a ballot contains too much information linked together. If there are r rows in the $3 \times r$ matrix of the multi-ballot, then there are $3r$ cells in the matrix, and a ballot contains and links together r individual cells.

We discuss three approaches for dealing with this attack: by ignoring it, by casting ballots by cells (individual bubbles) rather than by columns, and by having voters exchange receipts as they leave the poll site.

Determine if it is likely to be a problem

The first approach is to try to figure out more carefully, either with mathematical models or with simulations, whether or not this is a realistic concern. How likely is it that an adversary can actually figure out unambiguously how a voter voted, given the voter's receipt copy and the posted list of cast ballots?

This is not easy to work out, as it depends on how voters utilize their freedom to mark their ballots.

In some cases it can be a serious problem. If *all* voters completely fill in the first ballot, leave the third ballot completely blank, and give their choices on the second ballot, and take a copy of the second ballot home, then if there are four or more candidates in some race, the voter's retained copy can only be matched up with a completely-filled first-column ballot and a completely empty third-column ballot, thus revealing how each voter voted. This is a very worst-case example, since it requires that all voters vote this way.

With more arbitrary patterns of voting, one could crudely estimate that it might require the election to have a dozen races or more before an adversary can start matching up valid triples. Even then, it is very delicate, and there may be multiple ways in which a given ballot could participate in a valid triple.

One could attempt to argue that ThreeBallot provides an order of magnitude improvement in ensuring integrity, and so a small chance of eroding voter privacy is perhaps acceptable. (Certainly, those who, say, vote by mail are in potentially a much worse situation, since an adversary can force them to reveal their ballot before it is mailed.) But we should first see if we might not be able to do better here. The following two sections give some other approaches to handling this problem.

A cell-based approach

Our second approach provides a solid "fix" to the reconstruction problem, but at the cost of making the whole scheme exceptionally awkward and difficult. However, it demonstrates that the problem is fixable in principle, so the quest is then to find the best solution.

The idea is to change the focus from ballots as columns of an $r \times 3$ array, to ballots as single cells or entries of that array. Each entry would have its own "cell ID." The voter casts $3r$ ballots, one for each cell. As a receipt, the voter retains copies of r cells, one from each row, selected arbitrarily.

This makes the method much(!) less practical, but absolutely defeats any reconstruction attack. There is no way that an adversary can look at a collection of $3nr$ individual "cell ballots", and figure out the vote corresponding to any given receipt of r ballots. There will be multiple cell receipts for each row, some being unfilled and some being filled; they can be grouped into triples in a huge number of ways, and there is no way to tell which way is correct.

ThreeBallot with Exchanged Receipts

However, the best approach to the problem may derive from thinking about the functions of the receipts a bit more carefully.

The receipt is used in two ways: it is compared by the voter against its corresponding ballot before the ballot is cast, and it is used when the voter checks to see that the bulletin board contains a corresponding ballot.

The voter should check that her receipt actually matches the corresponding ballot that she will be casting, before she casts her three ballots. Only the voter can do this check, and she needs to have her own receipt and her own ballot in hand to do this comparison.

The reconstruction attack only works for an adversary, however, if the voter is known to be bringing a copy of *her own ballot* home as a receipt.

On the other hand, for the purpose of integrity checking the contents of the bulletin board, the voter only needs to bring home a copy of *some ballot that was actually cast.* If the receipt brought home by the voter isn't necessarily a copy of one of her own ballots, then there is no way for an adversary to bribe or coerce her, even with her cooperation and even if the adversary is able to successfully link together ballot triples on the bulletin board.

Here are some ideas for ensuring that an adversary can't count on the receipt brought home by a voter necessarily being a copy of one of her actual ballots:

- Ensure that voters have a mandatory opportunity to exchange receipts on their way out of the poll site.

- Keep a basket near the exit door. At the the beginning of the day, "pre-load" the basket with six "dummy" receipts. The serial numbers of the dummies will be officially recorded and posted on the bulletin board. When the voter leaves, require her to take away one of the receipts in the exit basket, and then to leave her own receipt in the basket. The receipts remaining in the basket at the end of the day are officially recorded and posted on the bulletin board. (This protocol is due to Devegili [8], and is known as the "Farnel" protocol, although

Farnel proposed it for ballots and not receipts. See also Araújo et al. [4] for a description of Farnel in English and an adaption to electronic voting.)

- Voters could be given two copies of their selected ballot, and then could exchange one of them with one of someone else's receipts, or via an exit basket. (Indeed, once you have forced exchanges, you could give the voter a copy of each of her three ballots, assuming that she is likely to check their correctness, but that she won't be bringing all of them home; she'll be bringing home other legitimate copies of ballots instead.)

This last approach, of enabling or requiring exchanges of ballot copies, say by using the Farnel protocol, seems the best. I'm optimistic that it can be implemented in such a way as to prevent an adversary from effectively bribing or coercing voters, even if the adversary could figure out some valid triples of ballots from the bulletin board.

Aside: OneBallot with Exchanged Receipts

Such "receipt exchange" protocols could also plausibly be used with a conventional opscan approach ("OneBallot"), where the voter receives a copy of her (single) ballot, checks that it correctly corresponds to this single ballot she will be casting, casts her single ballot, then (potentially) exchanges her receipt for another legitimate receipt before leaving for home. All cast ballots are posted on the bulletin board as usual, and the voter can check that the receipt she has indeed appears on the bulletin board, even though it may not be a copy of her own ballot. We can call this approach "OneBallot with Exchanged Receipts"; and because it is simpler than ThreeBallot with Exchanged Receipts, it is worth closer examination. Such receipt-exchange protocols can be modelled on the paper ballot exchange protocols of Farnel [8] (see also [4]), as sketched already above.

6 Usability

6.1 Usability for Voters

The ThreeBallot voting process is more complex than current conventional voting systems, so the impact on usability must be considered.

Of course, the main method for making sure that the voting system works well for voters is voter education. Although ThreeBallot is new, it is not very complicated, and a little voter education should make its operation clear.

However, if a voter makes a mistake, the process of recasting her ballot is not so simple. (Well, it's like opscan: you need to start over with a clean ballot.)

Voters who have difficulty with filling out a ThreeBallot multiballot could be given simplified instructions (e.g. always fill in the bubbles from left to right in a row).

Or, voters could be given preprinted ballots, where each preprinted ballot already has one randomly-placed mark in each row. Voters would then be told to place an additional mark in the row of their chosen candidate.

Or, a voter could use a conventional ("OneBallot") methods, since you can "mix" OneBallot and ThreeBallot ballots together (see Section 9.4).

Still, any additional increase in the complexity of voting is certain to cause some voter confusion and problems, so there is certainly a potential price to be paid, in terms of usability, for the security benefits of ThreeBallot.

ThreeBallot could be implemented with electronic ballot printers (see Section 9.3) that print out a multi-ballot or three ballots at once. Using electronic ballot printers is also a typical way of making voting systems more accessible.

The usability of ThreeBallot can be improved and refined with experience, and voters would become familiar with it over time. Perhaps voters would enjoy being able to vote "more than once"!

The benefit is that ThreeBallot provides major improvements in election integrity, at the cost of some impact on the ease of voting.

6.2 Usability for Pollworkers and Election Officials

ThreeBallot does make some extra work for pollworkers, since the number of paper ballots cast that need to be handled is now three times as large as with conventional ("OneBallot") voting.

The benefit, of course, is that voters may now enjoy a higher degree of confidence in the integrity of the voting process and in the election results.

7 Other considerations

In this section, we list a number of other concerns and considerations (most of which have been proposed by readers of earlier versions of this paper—thanks!).

Inefficient use of ballot page

The ThreeBallot format can be criticized as making inefficient use of the ballot page, compared say to traditional opscan layouts, which can use multiple columns to handle multiple races. ThreeBallot may thus require more ballot pages (which may lead to voter confusion).

Dependence on the checker

If the checker becomes inoperable, there is no way for voters to proceed to vote.

Fleeing voters

Voters are known to "escape" with ballots. With ThreeBallot, a voter should not be allowed to escape with just *some* of her ballots—she must call them all or cast none of them. Otherwise, she is essentially discarding a ballot, and could do so in a way that gives her vote undue influence.

8 Discussion

Cryptographic techniques can also provide all of the security properties of ThreeBallot. See Chaum [5], Chaum et al. [6], Ryan et al. [18, 17], Karloff et al. [13], Smith [22], and Adida [3] for presentations and discussions of cryptographic voting methods.

However, ThreeBallot achieves very nearly the same security properties, without the use of cryptography. (ThreeBallot's resistance for vote-buying is weaker, however.)

(I note for the record here that I have nothing against cryptographic voting methods—indeed, I find them very appealing. They are sometimes criticized, however, as being difficult or impossible for the average person to understand. (To be fair, the average person doesn't really understand software and its security risks either!) Thus, it is of interest to see to what extent the security properties of cryptographic schemes, such as "end-to-end verifiability," can be achieved in a "low-tech" manner, without using cryptography at all. The current paper is an effort in this direction; it may also be a useful starting point of study for those interested in cryptographic voting methods, as it embodies many similar principles.)

Recovery from some errors (e.g. too many ballots on the bulletin board) can be problematic, but not more so than with paper ballots today, in general. ThreeBallot is really just a paper ballot scheme, with the usual issues and remedies, except that voters cast three ballots that were constructed in a novel manner. Rescanning the cast paper ballots may suffice to fix many of these problems.

ThreeBallot also has pedagogic value for explaining various aspects of verification (e.g. verifying that one's vote was cast as intended versus verifying that one's vote is actually counted as cast versus verifying that the computed tally is correct), and in explaining by contrast various cryptographic voting protocols, which are quite similar to ThreeBallot in overall structure, but which use different detailed mechanisms.

Note that the voter is getting not only verification that her vote is "cast as intended" (as with most VVPAT or paper-trail systems), but also getting evidence that her vote is actually affecting the final tally as it should.

So, the ThreeBallot voting system seems to give a nice level of end-to-end verifiability with "plausible" (but not great) user interface, without using any cryptography.

One minor point regarding ThreeBallot: although all ballots are posted on the bulletin board, political scientists clearly won't find them as interesting or useful as they would find "real ballots" (OneBallot ballots).

9 Variations and Improvements

We have thus far presented and discussed the main architectural components of a ThreeBallot voting scheme: vote-by-rows but cast-by-columns, take a column copy home as a receipt, and post all ballots on a public bulletin board. We have also already discussed a number of refinements and details, having to do with ballot ID's, checking legality of multi-ballots, exchanging receipts, etc. We now review a number of further variations and extensions. I'm sure that there are nonetheless many more ways to implement the ThreeBallot architecture than are presented here!

9.1 Multi-ballot formats

The multi-ballot could be implemented in a number of different ways, in addition to the two that have been discussed already (three identical columns or three identical separate ballots). For example, you could have a first column with the candidate names, and then three following columns with just bubbles to be filled in. These last three columns are then the ballots; the first column can be discarded, as the candidate names should be in standard order.

9.2 Using existing opscan ballots

One could implement ThreeBallot using current opscan ballots. The voter fills out three complete ballots, and submits all three, keeping a copy of one. They all need unique serial numbers, and the checker needs to scan all three in order to do its thing. (Indeed, there is no reason why the three ballots need to be on the same piece of paper, as originally indicated, except for voter convenience when filling them out...)

9.3 Using a DRE with ThreeBallot

An interesting way to implement ThreeBallot is to use an EBM or EBP (electronic ballot marker or electronic ballot printer). You fill out your choices on the touch screen, and it prints out the ThreeBallot multi-ballot (or perhaps three single ballots), plus the copy of one ballot. (It could also print out a copy of each ballot, as long as the voter is required to destroy two of the three copies.)

However, the the voter still has to verify that the printed ThreeBallot multi-ballot accurately reflects her intentions.

Such an approach also then makes voting more accessible.

9.4 Mixing ThreeBallot with OneBallot

A conventional opscan voting system might be called OneBallot—each voter votes just once and can't take away a copy of her vote cast, whereas with ThreeBallot the voter cast three ballots, and takes away a copy of (an arbitrarily and secretly chosen) one of those three ballots.

You can actually mix these two systems. A OneBallot voter can toss her ballot in the same ballot box that a ThreeBallot voter places her three ballots in. The public bulletin board should indicate for each voter whether she is a OneBallot voter or a ThreeBallot voter, so that anyone may check that the bulletin board contains the correct number of ballots.

This provides a transitional path from OneBallot voting to ThreeBallot voting, as voters can choose which system to use.

Counting is the same, as each voter contributes at most one additional vote to their selected candidate (compared to the competing candidates).

One nice feature of mixing the two systems is that the OneBallot ballots are protected by being in the same ballot box as the ThreeBallot ballots, since an adversary will be deterred from deleting or modifying ballots since they might be ThreeBallot ballots which voters have retained copies of. (OneBallot ballots must be valid in the usual sense, without overvotes or undervotes. But a ThreeBallot ballot might also be valid in that sense, so an adversary would be deterred from deleting or modifying just ballots that are valid OneBallot ballots. Moreover, if OneBallot voters are also keeping receipts, as they might if they are participating in Receipt Exchanges (Farnel protocol [8]), then the adversary is further deterred.)

9.5 Write-In Votes

Since the voting scheme is essentially the same as with op-scan, you can handle write-ins in the same way.

The voter must make a mark to indicate that she wishes to specify a write-in candidate. (She does this on exactly two of her three ballots.) On the following blank line, she writes in the name of her desired candidate.

The checker should check that the same write-in candidate is given in both of her selected ballots (I can imagine that this check could be omitted with little harm, since the voter is then just splitting her vote. But this is probably illegal in most states.) The checker should also check that the write-in name is different than that of the already listed candidate names.

The scanner can either do OCR on the write-in name, or capture a bit-map of the written name. (There are the obvious concerns that a voter may try to make her vote identifiable here, so additional mechanisms may be needed to handle write-in votes.)

9.6 Combining the checker with the scanner

As presented, there are two scanning operations in ThreeBallot: the multi-ballot is first scanned in the checker, and then later the individual ballots in the ballot box are scanned in order to be placed on the bulletin board.

Can (or should?) these operations be combined? How about a single device that not only checks the legality of the multi-ballot and separates it into ballots, and which produces a receipt for the voter, but which also produces an electronic set of records for the bulletin board?

While this may be attractive from an economy point of view, it introduces a plethora of security concerns.

I don't think such an approach would be acceptable unless the machine at least actually printed out three receipts, one for each ballot, so the machine wouldn't know which ballots were going to be checkable by the voter. The voter might then choose to keep one such receipt, and discard the others. Or, the voter might keep them all, but participate in a procedure for Exchanged Receipts.

I would anyway want to see a strong procedure for Exchanged Receipts used, since the machine knows in principle which triples of ballots are linked (even though it is not supposed to remember this). (But as noted earlier, if we can really develop strong procedures for Exchanging Receipts, perhaps we can just use it on a OneBallot voting method, rather than on ThreeBallot?)

While this direction doesn't seem necessarily unworkable, it isn't obviously workable either; further elaboration study is needed to see if a reasonable level of security could be achieved.

9.7 More than Three Ballots

The scheme can easily be generalized to use more than three ballots. For example, a voter could use five ballots, and mark four to indicate "**FOR**" and three to indicate "**AGAINST**".

(In general there just needs to be a number d such that voting "**FOR**" requires making d more marks in a row than voting "**AGAINST**", and such that completely marked or completely unmarked rows are illegal.)

There is no apparent advantage to using more than three ballots.

Using only two ballots doesn't work, as the two ballots would need to stay linked together in order for the tally to be counted properly. (Note that the only potentially usable function of the two ballots would be "exclusive-OR", if you want to preserve the fact that the retained ballot copy can contain an arbitrary pattern.)

9.8 Other Vote-Counting Methods

The ThreeBallot system works fine for "approval" voting, where each voter merely indicates, for each candidate, whether or not they approve of that candidate. (Effectively, here we are merely removing the "race constraint" that a voter may vote "**FOR**" at most one candidate.)

ThreeBallot can be modified to support "range voting" (see Smith [21]), wherein each voter expresses a "degree of approval" of each candidate. One could use more than three ballots. For example, you could have seven ballots, and require voters to make between one and six marks in each row. Or, you can modify ThreeBallot so that each "mark" is a number between 0 and 99, with the row constraints appropriately modified.

ThreeBallot does not work well (or at all) for vote-counting methods such as ranked preference voting systems (such as Borda counting, IRV (instant runoff voting), or the Condorcet method) wherein voters provide a list of all the candidates in their order of preference.

ThreeBallot also doesn't work for straight-ticket voting (where a vote for the party implies votes for all the party's candidates), or other ballot situations where there is potential interaction or contingencies between the ballot questions.

10 Conclusion

We have presented a new voting system, ThreeBallot, that provides a high degree of verifiability—voters can verify that their votes are cast as intended, and can check that their vote is included in the final tally. All cast ballots are published, and tampering with votes can be detected.

This is the first time such end-to-end verifiability has been obtained without the use of cryptographic techniques. The principles employed by ThreeBallot are simple and easy to understand.

However, ThreeBallot's resistance to vote-buying is not as strong as I had hoped; perhaps some improvement in this regard can be found.

I encourage the reader to send me feedback, and to develop further improvements, extensions, and implementations of ThreeBallot.*

Acknowledgments

I'd like to thank Ben Adida, Andrew Appel, Jessy Baker, Alan Bawden, David Chaum, Ronald Crane, Chris Crutchfield, Kathy Dopp, John Kelsey, Silvio Micali, Peter Neumann, Ben Riva, Alex Rivest, Julie Shamos, Michael Shamos, Emily Shen, Warren Smith, Charlie Strauss, and Dan Wallach for helpful feedback and comments. (Of course, the fact that they provided helpful comments does not necessarily mean that they agree with everything in this paper; I take full responsibility for any errors or oversights here.)

*(Just for the record: ThreeBallot is hereby placed in the public domain—I am not filing for any patents on this approach, and I encourage others who work on extensions, improvements, or variations of this approach to act similarly. Our democracy is too important...)

References

[1] Brennan Center For Justice Task Force on Voting System Security (Lawrence Norden, Chair) . The machinery of democracy: Protecting elections in an electronic world. Technical report, Brennan Center for Justice, 2006. Available at: `http://www.brennancenter.org/programs/dem_vr_hava_machineryofdemocracy.html`.

[2] ACCURATE. `http://accurate-voting.org/`.

[3] Ben Adida. *Advances in Cryptographic Voting Systems*. PhD thesis, MIT Department of EECS, August 2006. Available at: `http://crypto.csail.mit.edu/~cis/theses/adida-phd.pdf`.

[4] R. Araújo, R. Custódio, A. Wiesmaier, and T. Takagi. An electronic scheme for the Farnel paper-based voting protocol. In *ACNS'06*, 2006. Available at: `http://www.cdc.informatik.tu-darmstadt.de/~rsa/papers/eFarnel-ACNS2006.pdf`.

[5] David Chaum. Secret ballot receipts: True voter-verifiable elections. *IEEE J. Security and Privacy*, pages 38 – 47, Jan/Feb 2004.

[6] David Chaum, Peter Y. A. Ryan, and Steve A. Schneider. A practical, voter-verifiable election scheme. Technical Report CS-TR-880, University of Newcastle upon Tyne School of Computing Science, December 2004. Available at: `http://www.cs.ncl.ac.uk/research/pubs/trs/papers/880.pdf`.

[7] Election Assistance Commission. `http://www.eac.gov/`.

[8] A. J. Devegili. Farnel: Uma proposta de protocolo criptográfico para votação digital (in portuguese). Master's thesis, Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, Brasil, 2001.

[9] Andrew Gumbel. *Steal This Vote*. Nation Books, 2005.

[10] Douglas W. Jones. Voting and elections. `http://www.cs.uiowa.edu/~jones/voting/`.

[11] Douglas W. Jones. Counting mark-sense ballots: Relating technology, law, and common sense, 2002 (rev. 2003). Available at: http://www.cs.uiowa.edu/~jones/voting/optical/.

[12] Douglas W. Jones. Chain voting, August 26, 2005. Available at: http://vote.nist.gov/threats/papers/ChainVoting.pdf .

[13] Chris Karlof, Naveen Sastry, and David Wagner. Cryptographic voting protocols: A system perspective. In *Proceedings 14th USENIX Security Symposium*, August 2005. Available at: http://www.cs.berkeley.edu/~nks/papers/cryptovoting-usenix05.pdf.

[14] CalTech/MIT Voting Technology Project. http://www.vote.caltech.edu/.

[15] Brian Randell and Peter Y. A. Ryan. Voting technologies and trust. *IEEE Security and Privacy*, 4(5):50–56, September/October 2006.

[16] Ronald L. Rivest. Voting resources page. http://theory.csail.mit.edu/~rivest/voting/index.html.

[17] Peter Y. A. Ryan and Thea Peacock. Prêt à Voter: A system perspective. Technical Report CS-TR-929, University of Newcastle upon Tyne School of Computing Science, September 2005. Available at: http://www.cs.ncl.ac.uk/research/pubs/trs/papers/929.pdf.

[18] Peter Y. A. Ryan and Steve A. Schneider. Prêt à Voter with re-encryption mixes. Technical Report CS-TR-956, University of Newcastle upon Tyne School of Computing Science, April 2006. Available at: http://www.cs.ncl.ac.uk/research/pubs/trs/papers/956.pdf.

[19] Roy G. Saltman. *The History and Politics of Voting Technology: In Quest of Integrity and Public Confidence*. Palgrave Macmillan, 2006.

[20] Michael Ian Shamos. Electronic voting—evaluating the threat, March 1993. Presented at Third Conference on Computers, Freedom, and Privacy (Burlingame, California). Available at: http://www.cpsr.org/prevsite/conferences/cfp93/shamos.html.

[21] Warren D. Smith. Range voting: The best way to select a leader? http://rangevoting.org/SmithWM.html.

[22] Warren D. Smith. Cryptography meets voting, September 10 2005. Available at: http://www.math.temple.edu/~wds/homepage/cryptovot.pdf.