Three Voting Protocols: ThreeBallot, VAV, and Twin

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

Warren D. Smith
Center for Range Voting, 21 Shore Oaks Drive, Stony Brook NY 11790
warren.wds@gmail.com

Abstract

We present three new paper-based voting methods with interesting security properties. Our goal is to achieve the same security properties as recently proposed cryptographic voting protocols, but using only paper ballots and no cryptography. From a security viewpoint we get reasonably close, particularly for short ballots. However, our proposals should probably be considered as more "academic" than "practical."

In these proposals, not only can each voter verify that her vote is recorded as intended, but she gets a "receipt" she can take home that can be used later to verify that her vote is actually included in the final tally. But her receipt does not allow her to prove to anyone else how she voted. All ballots cast are scanned and published in plaintext on a "public bulletin board" (web site), so anyone may correctly compute the election result.

In ThreeBallot, each voter casts *three* paper ballots, with certain restrictions on how they may be filled out. These paper ballots are of course "voter-verifiable."

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.

VAV is like ThreeBallot, except that the ballot-marking rules are different: one ballot may "cancel" another (VAV = Vote/Anti-Vote/Vote). VAV is better suited to – i.e. yields better security properties for – Plurality and preference (Borda, Condorcet, IRV) voting, while ThreeBallot is better suited for Approval and Range voting.

Finally, we introduce "Floating Receipts," wherein voters may take home a copy of *another voter's* ballot. (She doesn't know whose ballot, though.) Floating Receipts are well-tuned to the security requirements of ThreeBallot-like schemes, and we examine protocols for achieving them.

Our final voting system, Twin, is based almost entirely on Floating Receipts. Each voter casts a single ballot and takes home a single receipt. Twin is quite simple and close to being practical.

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 proposals presented here are an attempt to satisfy these constraints *without* cryptography. We get pretty close.

As in the cryptographic proposals, the proposals presented here use a "public bulletin board" (PBB) – a public web site where election officials post copies (but now in plaintext) of all of the cast ballots and a separate list of the names of the voters who voted. (Some states might post voter ID's rather than voter names.)

ThreeBallot, VAV, and Twin provide 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.

*The latest version of this paper is always at http://people.csail.mit.edu/rivest/RivestSmith-ThreeVotingProtocolsThreeBallotVAVAndTwin.pdf

1.1 Background on voting

The following books [26, 12], reports [1], theses [2], articles [21], and web sites (Jones[1], Rivest[2], CalTech-MIT[3], ACCURATE[4], EAC[5]) are recommended.

1.2 Some important single-winner voting systems [18, 20]

Our protocols have different levels of compatibility with some of the important single-winner voting systems. So we summarize the latter here.

Plurality: Your vote is the name of one candidate. Most-named candidate wins.

Approval voting [7]: Your vote is the set of candidates you "approve." Most-approved candidate wins.

Borda voting [25]: Your vote is a rank-ordering of the C candidates. A candidate receives $C - K$ points for being ranked Kth on a ballot. Candidate with the greatest point total ("Borda count") wins.

Condorcet systems: Votes are rank-orderings. If a "Condorcet winner" candidate exists that is preferred pairwise over each opponent by a majority of the ballots, then he wins. Otherwise, there are a number of inequivalent techniques that have been suggested to determine a winner, and it shall not matter to us which one, provided it depends only on the $C \times C$ matrix of pairwise counts.

"Instant runoff" (IRV): Votes are rank-orderings. The candidate top-ranked on the fewest ballots is eliminated (from the election, and from all ballots), reducing it to a $(C-1)$-candidate election, and the process continues until only one candidate remains – the winner.

Range Voting[6]: Your vote for each candidate is an integer in some fixed range (e.g. $0 \ldots 9$ for "single digit range voting"). The candidate with the greatest total score wins. Approval voting is just range voting with integer range $[0, 1]$.

Our secure-voting protocols do the following jobs. ThreeBallot is intended to handle approval and range voting. VAV and Twin can handle *any* voting system, including ones we have not described such as multi-winner Hare/Droop reweighted STV.

[1]Douglas W. Jones. Voting and Elections. `http://www.cs.iowa.edu/~jones/voting/`

[2]Ronald L. Rivest. Voting resources page. `http://people.csail.mit.edu/rivest/voting`

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

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

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

[6]The Center For Range Voting. `http://RangeVoting.org`

BALLOT		BALLOT		BALLOT	
Abe	•	Abe	•	Abe	○
Bob	○	Bob	•	Bob	•
Cal	•	Cal	○	Cal	○
Dik	○	Dik	○	Dik	•
r9>k*@0e!4$%		*t3]a&;nzs^_=		u)/+8c$@.?(	

Figure 1: In this *Approval-voting* multiballot, the Three-Ballot voter is approving Abe and Bob while disapproving Cal and Dik. Note coded ballot IDs at bottom.

2 ThreeBallot

We now describe ThreeBallot in more detail. (An earlier version of ThreeBallot was posted [22] in 2006; it contains some variations and discussion not included here for lack of space, but unfortunately also some flaws.)

2.1 Voting in ThreeBallot

Checking In at the Poll Site

The voter receives a paper "multi-ballot" to vote with. A multi-ballot consists of three paper ballots, printed on separate sheets. (They could be printed on a single sheet, with perforations for later separation, but this introduces some unnecessary security concerns, so we do not recommend this approach.)

We imagine that the poll-site has a bin of pre-printed blank ballots, and that the voter randomly selects three to form her multi-ballot.

The Multi-Ballot

ThreeBallot is perhaps most easily viewed as an extension of "mark-sense" ("optical scan" or "opscan") systems [13].

Each ballot in the multiballot is an identical complete ballot; think of it as a standard opscan ballot with the addition of a unique coded ballot ID on the bottom of each ballot. See Figure 1. The upper voting region lists candidate names with matching opscan bubbles that can be filled in (marked).

Each ballot ID is different from other ballot ID's on the multi-ballot or elsewhere. The ballot ID's on a multi-ballot are thus unrelated, random (but unique) ballot ID's.

The ballot ID might also be a long random string of symbols or some other unique identifier. For now, we assume that coded ballot ID's are pre-printed on the ballots, but we'll see there can be security advantages to having ballot ID's added later instead by the voter or by the "checker" (see §2.1).

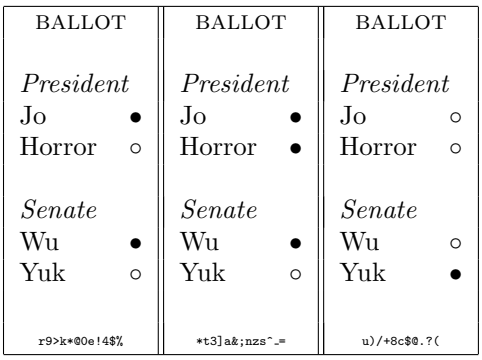

Figure 2: A multiballot bundling two races (President and Senator). This ThreeBallot voter is plurality-voting for Jo for President and Wu for Senator. (There are many other equivalent ballots.)

Filling Out The Multi-Ballot

For simplicity, we assume for now that we are using *approval voting*, the simplest form of voting, where the voter merely indicates for each candidate whether or not she[7] approves of that candidate (see Brams et al. [7]). This is like ordinary plurality voting except that voters may approve of more than one candidate in a race. Figure 1 gives an example of a filled-out multi-ballot.

We will call a filled-in bubble a *mark* (for a candidate).

The voter is given the following instructions for filling out the multi-ballot.

- You have three ballots; you will be casting all three. Proceed candidate by candidate through the multi-ballot. Each candidate has three "bubbles," one on each ballot.
- To vote **for** (approve) a candidate, fill in any *two*, but exactly two, of the bubbles for that candidate.
- To vote **against** (disapprove) a candidate fill in any *one*, but exactly one, of the bubbles for that candidate.
- Your multi-ballot *will not be accepted* if any candidate has no bubbles filled or has three bubbles filled.
- You may vote **for** (approve) any number of candidates, including none of them or all of them.

Checking the Filled-Out Multi-Ballot

When a voter has marked her choices, she inserts her multi-ballot (i.e, her three ballots) into a "checker machine," to check the validity of her multi-ballot. The checker 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 (these are "row-constraints").

If the multi-ballot is invalid, the checker beeps, indicates where the errors are, and gives the ballots back.

Getting a Ballot Copy as a "Receipt"

If the multi-ballot is OK, the checker beeps (nicely), and asks the voter to choose *one* of the three ballots (which should be visible under glass), e.g. by pushing button "1," "2," or "3." The machine then makes a *copy* of the selected one and gives it to the voter as her take-home "receipt". It is important that the voter chooses *secretly* and *arbitrarily* which ballot to copy. Which button she pushed should be known only to her and *not* "remembered" by the machine. The machine, indeed, should be "dumb" and not capable of remembering anything about the ballots. This receipt should be printed on different paper so it looks different than her original ballots, and it should be difficult to alter or forge. The security of the receipt could be enhanced even further by "certifying" it by printing on it a "digital signature" that could have only come from an official, government voting machine.

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

Casting Three Ballots

The machine now drops all three original ballots into the ballot box, in view of the voter. It ensures that the voter either cast all three ballots, or none.

The ballot box has the usual property that it scrambles the ballot order, destroying any indication of which triple of ballots originally went together, and what order ballots were cast in.

The voter then signs her name on her entry on the registration list to indicate that she voted—in such a way that her three ballots *only* enter the ballot box, and she *only* gets her receipt, if and when she signs her name off on that list.

Going Home

The voter takes her receipt, and goes home. (See §4 for a possible extension to "Floating Receipts", where the voter will take home a copy of some other voter's ballot instead of her own.)

Posting the Ballots

At the end of election day, all cast ballots are scanned and published on the PBB. (Cast ballots

[7] For expository convenience, the voters will be female, and everybody else male.

should *not* be scanned and posted as they are cast, but only at the end of election day.)

Here scanning produces a "compact" representation of the voter's choices that just records the marks present and the ballot IDs. A pixel-level scan is *not* used because a voter might mark the margin of the ballot so she could identify her ballot image later.

The election officials also separately post a list of the names of all voters who voted.

Checking the Integrity of the PBB

Once home, a voter may check that her receipt matches a ballot posted on the PBB. We note that ballot ID's are coded to make them difficult to remember at the poll-site; but they can be coded with symbol strings that are still easy to type into a web site. (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 or is too busy to access the PBB herself. Adida [2] discusses helper organizations. The helper organization should wait until election day is over and all ballots are posted before looking at the receipts received, so that it can not gain any possible statistical information on the voting trends during the day.)

If the PBB doesn't contain a matching ballot, she takes her receipt to an election official within two days and files a protest. The election official may examine the voter's receipt to determine its authenticity, and may authorize a rescan of the cast paper ballots.

Tallying and Announcing the Winners

The ballots can be tallied by anyone, since they are publicly posted in plaintext on the PBB. No decryption is needed.

The winners can be announced.

Each voter has marked once or twice for each candidate; the checker has enforced this.

So each candidate's tally will be "as usual," except that each total is inflated by the number n of voters. The election outcome is the same. For example, if candidates A, B, and C would ordinarily have received a, b, and c votes, respectively, then with ThreeBallot the final tallies will be $n+a$, $n+b$, and $n+c$; the true vote totals are obtained by subtracting n from the total number of marks for each candidate.

What about range voting?

For a Range rather than Approval voting election: same procedure works, except for these changes. Suppose the allowed scores are $\{0, 1, 2, \ldots, R\}$, where, e.g. $R = 9$ for single-digit range voting. Instead of marking (or not) a bubble to indicate approval or disapproval of that candidate the voter fills in a slot with a *number* in $\{0, 1, 2, \ldots, R\}$ (or by marking one of $R+1$ bubbles). The candidate constraints are that the sum S of the candidate's three scores satisfy $R \leq S \leq 2R$. The checker enforces this. To give Perot score P, $0 \leq P \leq R$, she should make sure $S = R + P$. See Figure 3.

BALLOT		BALLOT		BALLOT	
Xerxes	9	Xerxes	0	Xerxes	0
Yu	5	Yu	9	Yu	4
Zippy	4	Zippy	1	Zippy	5
r9>k*@0e!4$%		*t3]a&;nzs^_=		u)/+8c$@.?(	

Figure 3: This single-digit *range voting* multi-ballot rates Xerxes=0 (worst), Yu=9 (best), and Zippy=1 (just above worst) (these are total scores minus 9).

This completes our description of the operation of the ThreeBallot voting system for Approval and Range voting.

2.2 Security Assumptions & Mechanisms

ThreeBallot has some unique concerns among voting systems, since the voter has as a receipt a copy of part of her vote, in plaintext, and all ballots cast are available in plaintext on the PBB.

Without some additional assumptions or mechanisms, the voter privacy of ThreeBallot can be attacked, as noted below. We now propose one assumption and one mechanism.

The **Short Ballot Assumption** (SBA) assumes that the ballot is short—there are many more voters in an election than ways to fill out an individual ballot—i.e., there are only a few races and only a few candidates in each race. That is, if the ballot has C candidates total, then (for approval voting) the number 2^C of voting patterns must be much smaller than the number of voters. Depending on the voting system used, the SBA may be more or less reasonable; for example a rank-order voting system with C candidates has $C!$ possible voting patterns, which may easily exceed the number of voters. See Table 1. It is reasonable to assume under the SBA that each possible ballot is likely to be cast by several voters. The security of all three of our secure voting protocols depends on the SBA.

An important related idea is *debundling* – separating different races, or even different candidates in the same race onto separate ballots, so that the SBA holds for each ballot.

With Approval and Range voting, the voter provides a *rating* for each candidate – "ap-

system	#patterns
Approval	2^C but only 2 if debundled
1-digit Range	10^C but only 10 debundled
2-digit Range	100^C but 100 debundled
Borda, Cndrct, IRV	$C!$
Plurality	C (VAV) but 2^C (ThreeBallot)

Table 1: Pattern-counts in different voting systems.

prove"/"disapprove," or a numerical score in, say, $[0, 9]$. Her rating for Xerxes is *independent* of her rating for Yu. So in these voting systems it is *unnecessary* to force her to rate all the candidates in a race on the same ballot. With Range and Approval voting, ballots can be debundled all the way down to *single candidates* if desired, thus guaranteeing that the SBA holds.

However, for Plurality, all candidates *must* be on the same ballot, because votes for the different candidates are *dependent*—if you vote for Xerxes, you may not vote for Yu. A plurality race with many candidates may have to violate SBA. (Well, you can have a multi-page ballot with separate coded ID on each one, as long as they are checked simultaneously. Long candidate lists cause problems for many voting systems.)

There is a clear SBA/bundling trade-off. Debundling would have been essential to achieve SBA for approval or range voting for the 2006 Congo presidential election, which had $C = 33$ candidates. But VAV (see §3) *could* handle Plurality for this election with no debundling; VAV can also handle rank-order ballot elections when $C!$ is well below the number of voters. In many cases no debundling may be needed.

"**Floating Receipts**" is a new security mechanism introduced in this paper (see §4), wherein a voter may bring home a receipt that is a copy of *some other* voter's ballot, and/or extra copies of receipts could be "floating" around in unknown hands. That can provide additional helpful layers of anonymity or security.

Twin depends on floating receipts, while VAV and ThreeBallot do not, *but* one can *add* floating receipts as an *extra security feature* to either, so that these protocols become secure even against a wide class of *collusive* attacks (§2.3.8). For simplicity we describe ThreeBallot and VAV without such collusion-protection, then will describe ways to add it in §4.

2.3 ThreeBallot Security — Integrity

The two main voting system security requirements are integrity of election results and voter privacy. We begin with integrity.

The voter can check that

- a ballot matching her receipt is posted on the PBB in the list of cast ballots (using the coded ID as the lookup key),
- the total number of ballots on the PBB is three times the number of voters who voted (the list of voters who actually voted is also published on the PBB).

The first check has no analogue in most current voting systems. We'll see these checks allow detection of several kinds of fraud. Of course, one has to ensure that the new security mechanisms can't themselves be easily attacked.

2.3.1 Adding Ballots can be Detected

An adversary can't add ballots to the PBB without putting more voter names on the PBB, which makes the fraud detectable. (Grandma, did you really vote? Weren't you sick that day?) On this issue of ballot stuffing, ThreeBallot is little different from other voting systems; the best defense is public oversight of voter check-in process and the posting of lists of voters who voted.

2.3.2 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 matches no ballot on the PBB. (The PBB has no ballot with the same ID, or shows one with different marks.)

Of course, an adversary might risk modifying just a few ballots, hoping to avoid detection since only 1/3 of the ballots are protected. It's key that *nobody knows which* 1/3, which assures that any large-scale fraud would get detected with even a low level of vigilance by voters or their proxies. (The actual chance of detecting such fraud can be computed with the usual sort of "auditing math" [5].)[8]

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

2.3.3 No Voter Coercion or Vote Selling

A design goal of the ThreeBallot system is that the voter should not be able to sell her vote, since her

[8] E.g.: to detect a *fixed* level of fraud (e.g. 6%) with significant probability (e.g. 95%), only a *fixed* number of voters (in this case 50) need to check, no matter how large the electorate.

receipt doesn't provide reliable information on how she voted.

Note how ThreeBallot follows the philosophy of "vote by rows, cast by columns"—viewing each candidate's votes as 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 whether the voter approved or disapproved any given candidate.

No matter whom she votes for, her receipt can have any possible pattern of marks. Moreover, the voter has complete control over what pattern is 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 express her desired voting preference and "outvote" ballot 1 as necessary. She then copies ballot 1 as her receipt to give the coercer.

2.3.4 The "Three-Pattern" Attack

There is another attack with which a coercer may attempt to buy votes or influence a voter's voting behavior.

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 PBB, the voter doesn't get paid (or is punished).

The attack fails if the Short Ballot Assumption holds, since each possible ballot pattern is likely to occur, many times.

2.3.5 Recounts and Audits

Because the ballots are paper, 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 required if enough voters credibly claim that their receipts aren't represented correctly or at all on the PBB.

2.3.6 Detecting Malicious Voters

The receipt may need some additional authentication (cf. Adida [2, §5.3]) to prevent voters from maliciously claiming that their (fabricated) receipt doesn't match any ballot on the PBB. This authentication could be a seal or sticker on the receipt, or (better) an unforgeable digital signature for {the vote on the receipt, the election-name ("2008 Presidential"), and the ballot ID}.

It is OK for the voter to let an official sign her receipt as an officially approved receipt, since voter privacy isn't threatened. However the digital-signing machine must *not remember* the ballot ID. This worrisome requirement can be avoided by not revealing the ID to the signer and not including it as part of the signed bit-string. In that case the digital signature would protect only the vote while the ballot ID would be protected by old-fashioned paper-and-ink antiforgery technology.

2.3.7 Attacking the Checker

The checker needs to be tested carefully. A maliciously modified checker could, e.g. allow some voters, by violating the row-constraints, to effectively have *three* votes!

Note that such illegal voting patterns can't be detected later, since the row-constraint can't be retested once the multi-ballot is split up. (Of course, in some cases you may be able to tell that such an attack has been mounted: e.g., if a candidate ends up with more than $2n$ marks.)

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 candidate conditions are exceptionally simple to check, and a simple hard-wired row-constraint checker may be sufficiently trustable that one can have confidence in its correct operation on election day.

One may compare this situation with other forms of VVPR (voter-verified paper records), such as DRE-VVPAT or ordinary opscan. ThreeBallot is like them in that the voter can directly verify her own paper ballot, to ensure that her intent has been captured correctly on paper. As with other VVPR methods we must worry about "casting multiple votes," and "ballot box stuffing." But ThreeBallot has an additional risk: that a corrupt checker would allow some voters to cast a "heavier" vote than others. But ThreeBallot allows voters to detect modification of the collection of cast ballots, whereas other VVPR schemes don't even attempt this. On balance ThreeBallot addresses better the more serious threats.

To compare this situation with that for 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.

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

2.3.8 Paying for Receipts

The adversary may be able to buy voters' receipts as they leave, *and* then be able to manipulate the contents of the PBB. (This is a "collusive attack," §4.1 since the vote-buyer and the PBB staff must collude for it to work.)

Knowing that the voter has given up her ability to contest PBB corruption, the adversary alters the PBB copy of her corresponding ballot. (The "floating receipts" approach (§4.1) defeats this attack by enabling multiple copies of a ballot as receipts.)

Voters should thus be cautioned not to casually discard or give away (the only copy of) their receipts. If she uses a "helper organization" (e.g. the ACLU) as a proxy to check PBB integrity, the voter might give the helper organization a *copy* of her receipt, rather than the original. (The receipt might be signed at the poll site with a bar-coded digital signature, so a copy is as "authentic" as the original here.)

There should also be strong safeguards on PBB modification for a "layered defense." Current voting systems rely for their security *entirely* on such safeguards, so ThreeBallot can't help but be an improvement.

This attack works for many cryptographic schemes in the literature; the only prior countermeasures we know of are by Ryan and Peacock [23, §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. [16, §5.2], who suggest voter education.

2.3.9 Chain Voting

In the *chain voting* [14] attack on paper-based voting systems, a buyer hands a voter a pre-marked ballot. She casts it as "her" vote, then gives the blank ballot she should have used to the vote-buyer (receiving payment and allowing the cycle to begin anew). The usual remedy ensures that a voter casts the ballot she was given. Ballots have tear-off stubs. When the voter first picks up a blank ballot, a random number (e.g. from dice) is written on the stub and recorded under that voter's name on a list; when she casts her ballot the stub number is checked, and the stub torn off and visibly destroyed. Poll procedures ensure voters cannot leave then re-enter the polls with ballots.

2.4 Voter Privacy in ThreeBallot

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

The first of Professor Michael Shamos's "Commandments" [27] on voting is:

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

A voter should not be able to violate her own privacy, even if she wishes: she should be unable to convince anyone else that she voted in a particular way. Otherwise she could sell her vote. (This is why we 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 could the voter show an adversary to persuade the adversary of how she voted? Three sorts of evidence are available:

- physical evidence she 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 cast ballots on the PBB.

2.4.1 Can Receipt violate Voter Privacy?

Can the voter violate her own privacy using her receipt?

Nothing prevents a voter from voting entirely honestly on her first ballot. If she then copies that first ballot for her receipt, it indicates exactly how her votes will be tallied.

But this receipt is at best a "reminder" of how she voted, not a proof that will convince anyone else as to how she voted. She is unable to intentionally violate her own privacy by showing someone else her receipt. (We have already argued, in §2.3.3, that a voter's receipt, by itself, bears no information about how a voter voted. So it cannot violate voter privacy.)

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

No one should be able to reliably and convincingly link together the three ballots on the PBB that con-

stitute an original cast multi-ballot. Otherwise, the voter's privacy is at risk.

If the multi-ballot were to be printed on a single sheet of paper (e.g., with perforations), then there would be a risk that the printer could remember which triples of ballot ID's were valid. (Some cryptographic voting schemes, such as Prêt à Voter [23] also have have security vulnerabilities at the ballot printers.) However, with our recommended procedure of having the voter randomly select three ballots from a bin to constitute her multiballot, this isn't a problem.

Risk Voter can identify her Multi-Ballot

The voter should not be able to record or remember the ballot ID numbers of her three ballots. (Voters should not be allowed to take photos of their multi-ballot! Cell-phones, video cameras etc. must be prohibited!)

We have five approaches for this:
1. Printing ballot ID's in a coded way, using punctuation and other symbols. Advantages: (a) simple. (b) hard for the voter to remember. Unfortunately it's easy for a voter to copy manually. **2.** Barcode. Somewhat harder to copy, but also harder to read later when looking the ballot up on the PBB.
3. ID's could be under "scratch-off." That's more secure, but precludes cheap and common printing equipment.
4. Ballot ID's could be printed using "visual cryptography" [19]; with this approach ballot IDs by themselves are just "random dots" which convey no information whatever. The voter is given the plastic overlay that allows her to read the ballot ID only *after* she receives her receipt. Almost impossible to copy or remember without a (incriminating) camera or overlay.
5. (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:

- All multi-ballots are initially identical and contain no ballot ID's.
- When the checker determines that a multi-ballot is OK, it prints three randomly generated ballot ID's on the three ballots, but retains the ballots for now.
- The voter selects which ballot she wants 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 match. 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 her ballot and unequal 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 or photographing them. It also complies with state laws (like California's) that require all blank ballots to be identical. We note that if its "random" number generator is deterministic or defective, voter privacy might be compromised.

Risk of Copying

It is better not to let the voter use a generic copying machine to make her receipt, lest she make a copy of all three ballots.

The procedures we've suggested are designed to ensure that the voter only gets a copy of one of her three ballots, and can't copy her other ballots or their ID's.

Reconstruction Attack

In a "reconstruction" attack the adversary examines all possible triples of ballots from the PBB, 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 taken together with either the ballot ID available on the voter's receipt, or if that voter made an agreement beforehand (like the three-pattern attack).

Under the Short Ballot Assumption, there will be many ways to piece ballots together, and the adversary gains insufficient information to combine triples of ballots to infer the voter's vote.

Strauss [31] provides some empirical evidence on the effectiveness of reconstruction attacks for various sized ballots, as do Jones et al. [15, 6]. Also see Cichon et al. [10] for some careful analysis of how short a ballot needs to be to provide voter privacy in ThreeBallot.

2.4.3 Early publication "threat"

Jeroen van de Graaf raised the concern that public PBB posts or "leaks" from helper organizations

might reveal statistical information about ThreeBallot vote totals "early" (say, halfway through the election); in contrast with full-cryptographic voting protocols. A quick fix is to require that the PBB and helper organizations not distribute partial information about their ballots until the polls close (§2.1).

But one of us (WDS) feels this really is less of a security issue than a voting system issue. Restricting partial information is effectively impossible, since exit polls may yield essentially the same information (or better), restricting early exit poll publication is probably unconstitutional, and nothing prevents anyone from privately commissioning exit polls whose results are made known only to the purchaser. The *real* problem is mainly that the Plurality voting method (§1.2) is highly vulnerable to insincere "strategic voting," whereas, e.g, in Approval and Range voting strategic decisions based on early returns have less importance.

2.5 ThreeBallot Usability

2.5.1 For Voters

The ThreeBallot voting process is more complex than current conventional voting systems, so the impact on usability must be considered. An interesting preliminary field study [15] indicates that ThreeBallot has significant usability issues.

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 nonetheless quite workable, and a little voter education may make its operation clear for many or most voters.

However, if a voter makes a mistake, the process of recasting a 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 for a candidate from left to right).

Probably the best approach is to merely let voters who have undue difficulty with ThreeBallot use conventional ("OneBallot") methods, since you can "mix" OneBallot and ThreeBallot ballots together (§2.6.1).

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

2.5.2 For Election Officials & Workers

ThreeBallot causes 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. Furthermore ThreeBallot may encourage inefficient use of the ballot page as compared to traditional opscan layouts, requiring even more ballot pages.

The benefit that compensates for this extra work and extra usability problems for voters, is, of course, a higher degree of confidence in the integrity of the voting process and election results.

2.6 Variations and Improvements

We have presented and discussed the main architectural components of a ThreeBallot voting scheme: vote-by-candidates (rows) but cast-by-columns, take a column copy home as a receipt, and post all ballots on a PBB. We now review further variations and extensions.

2.6.1 Mixing One- with ThreeBallot

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 casts three ballots, and takes away a copy of (an arbitrarily and secretly chosen) one.

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 in. The PBB should indicate for each *voter* whether she is a OneBallot voter or a ThreeBallot voter, so that anyone may check that the PBB contains the correct number of ballots, but it should not be possible to tell by examination of a *ballot* which type it was.

This provides a transitional path from OneBallot to ThreeBallot voting, as voters can choose which system to use. Our systems are compatible so counting is the same.

A nice feature of mixing the two systems is that the OneBallots are *protected* by being in the same ballot box as ThreeBallots, since an adversary will hesitate to corrupt any ballots as they might be ThreeBallots which voters have retained copies of. (OneBallots must be valid in the usual sense, e.g, for plurality voting, without overvotes or undervotes. But a ThreeBallot might also be valid that way, so an adversary would be prevented from deleting or modifying just OneBallot ballots. Anyhow, with approval voting all ballots are valid.)

2.6.2 Write-In Votes

ThreeBallot Range and Approval voting trivially permit "write-in" votes *if* voter-ratings of write-in candidates are handled debundled on *separate* ballots.

(If they were bundled on the same ballot as the regular candidates, voters could attempt to identify their ballots by writing-in unique "candidate" names. Cf. [17].) Range voting with high-precision scores also has to be forbidden for the same reason (insist on single-digit range voting).[9]

Our schemes for plurality and rank-order voting systems unfortunately cannot handle write-in candidates.

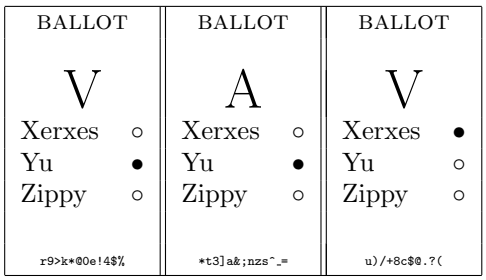

Figure 4: In this *Plurality-voting* multiballot, the VAV voter is voting for Xerxes. She here has chosen to make her *first two* ballots be the vote-antivote pair, and her *last* ballot is her real vote.

2.6.3 Other Vote-Counting Methods

As we have seen, ThreeBallot works fine for "approval" voting.

Although we prefer to handle plurality voting with VAV (§3), ThreeBallot can do it too; we add "race constraints" to the "row-constraints" already discussed; each race constraint permits the voter to approve at most one candidate per race. This makes the checker more complicated (it now must know which candidates are in the same race), but otherwise not much changes. (We note that an individual ballot might be part of a legal threeballot, yet now look like an illegal overvote; that is OK.)

ThreeBallot does not work at all for ranked preference voting systems such as Borda, IRV, or Condorcet. Our VAV protocol (§3) does.

2.7 ThreeBallot – Summary

Cryptographic techniques can also provide all of the security properties of ThreeBallot, and more. See Chaum [8], Chaum et al. [9], Ryan et al. [24, 23], Karloff et al. [16], Smith [30, 29, 28], and Adida [2] for presentations and discussions of cryptographic voting methods.

However, ThreeBallot achieves almost as good security properties, without cryptography.

We note for the record that we have nothing against cryptographic voting methods—they are very appealing, although a bit complex.

That's why this paper's goal is to see to what extent the security properties of cryptographic schemes can be achieved in a "low-tech" manner, without cryptography.

What happens after our protocols prove there was fraud? Election systems need a more graceful way to recover from errors than merely declaring failure; we somehow need to insert *accountability* and *corrective feedback*. Our protocols, while imperfect in that respect, seem as good, and in some ways superior, to present-day paper balloting. The potential for "superiority" is since in our schemes with *preprinted ballot IDs*, we can set up a chain of custody for all ballots within ballot-number intervals. E.g, there will be records saying "Joe Schnoz obtained custody of ballots 785400-823100 at 6pm on 30 September." Now suppose some cast ballots vanished or were altered, or fake voters voted (on stolen blank ballots). Then we *know* exactly who was supposed to have custody of the ballots with those numbers and when, so we can try to assign blame and stop the corruption.

Recovery from some errors (e.g. too many ballots on the PBB) can be problematic. Rescanning the cast paper ballots may suffice to fix many problems. ThreeBallot is really just a paper ballot scheme, with the usual issues and remedies, except that voters cast three ballots constructed in a novel manner, and have a new protocol for checking that at least one of their ballots is counted in the final ballot box.

ThreeBallot also has pedagogic value as a "stepping stone" when explaining cryptographic voting protocols, as it strives to achieve similar properties with simpler methods.

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 cryptography.

3 VAV

VAV stands for "Vote / Anti-Vote / Vote".

The VAV scheme is similar to ThreeBallot: each voter casts three ballots.

In VAV each ballot is pre-marked as either a "Vote" (positive, marked "V"), or an "Anti-Vote" (negative,

[9] Albeit in principle it could be handled by a further level of debundling – down below single candidates into single *digits* of large numbers.

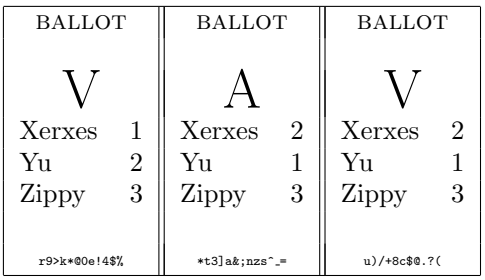

Figure 5: In this *Borda*, *Condorcet*, or *IRV* multiballot, the VAV voter is ranking Xerxes top, Yu second, and Zippy bottom. She here has chosen to make her *last two* ballots be the vote-antivote pair, and her *first* ballot is her real vote.

marked "A"). The voter casts one two Votes and one Anti-Vote, and we demand that one of the Votes exactly cancels the Anti-Vote (it is identical except for the A/V indicators). See Figure 4.

She takes home as her receipt a copy of any one of her three ballots, just as in ThreeBallot.

VAV can support *any* kind of voting system.

It is important that each ballot be *pre-labeled* as "V" or "A" and that the talliers explicitly eliminate each A-antivote and a matching V-vote before tallying. (One could imagine a VAV-like scheme without explicit VAV labeling, and while that would work for Condorcet and Borda[10] voting, it doesn't work for IRV.)

The only concern is that each Vote shouldn't contain too much information – that would allow the voter to cast a ballot highly likely to be unique, thus allowing her to sell her vote to a buyer who can verify its presence on the PBB. Thus, we are again assuming a version of the Short Ballot Assumption (SBA). The reason that VAV is superior to ThreeBallot for *plurality* voting, is that with VAV in a C-candidate race, the number of possible ballot-patterns is C – small, easily permitting satisfaction of the SBA – whereas with ThreeBallot it would have been 2^C – large, often forcing violation of the SBA.

VAV and OneBallot voting can also be mixed as in §2.6.1 (the OneBallots are all "V").

4 Floating Receipts

"Floating Receipts" is a simple and powerful security-enhancing idea. How can it help? ThreeBallot and VAV, as we have described them, can unfortunately be vulnerable to *collusive attacks* (e.g. modifying the PBB after obtaining some receipts) as described in §2.3.8.

The method proposed here ("floating receipts") addresses these concerns by having voters *take home copies of receipts other than their own.* Essentially, there is a big bin of receipts, and voters first make a copy (to take home) of one receipt already in the bin (leaving the original in the bin), then toss their own original receipt into the bin. This method has the following properties: **Anonymity:** No one knows which voter cast the ballot corresponding to a given receipt copy with any useful probability. **Exchange:** No voter takes home a copy of her own receipt. **Coverage:** A constant fraction of the original receipts are copied, with high probability (and there is no way for anybody to know what that subset is). **Collusion-Resistance:** An adversary has no efficient method for confidently obtaining *all* the copies of a given receipt.

The Exchange property fights any attacks wherein an adversary pays for receipts of a given form. Collusion-Resistance fights attacks where the adversary collects *all* copies of a given receipt, so as to evade PBB manipulation detection. Coverage ensures that any significant PBB manipulation has a good probability of being detected. And Anonymity is useful for use in voting schemes (like "Twin" below) that would otherwise not have anonymity.

Voters traditionally have been anonymous "going into" the voting process (submitting ballots). Floating receipts now provide a new layer of anonymization "coming out" (taking home receipt copies).

Note that to check the integrity of the PBB, the voter only needs to bring home a copy of *some* cast ballot. We admit she would have more motivation to check her *own* ballot, but assume that many voters will check anyway.

4.1 Making "floating receipts"

There are many ways of implementing floating receipts; we now elaborate the approach sketched above.[11]

Let T be a constant (e.g. $T = 20$), significantly less than the number of voters, such that $1/T$ is a "small" probability. There is an initially empty "bin" (receipt reservoir).

Each voter begins our "floating receipt" protocol having cast her ballots, and having the receipt. So far everything is as in ThreeBallot or VAV but we

[10] This was noted by Michael A. Rouse; you reverse order for the anti-vote.

[11] The method here, though developed independently, can be also usefully be viewed as an extension of the earlier "Farnel" protocol (see A.J. Devegili [11, 4, 3]) to handle ballot receipts rather than ballots themselves; in the Farnel protocol no take-home receipts were envisioned.

now consider adding an additional Floating Receipts layer of defense.

[Phase I] The first T voters put their original receipts in the bin, and receive nothing to take home.

[Phase II] After that every voter: (a) gets a *copy* to take home of a receipt chosen randomly from the bin (the original is returned to the bin), and then (b) deposits her own original receipt into the bin. The take-home receipt copies are certified as authentic as the voter leaves, with a notary-like stamp or a digital signature (as long as the signing device doesn't record what it sees). When the voter leaves she might also be required to sign the registration book to certify that she has voted.

After the polls close, the final contents of the bin may be discarded or published in some manner[12] according to the preferences of the election officials, or perhaps given to an election-monitoring organization—these receipts could also be checked against the PBB.

The voter may use her receipt copy to check the PBB integrity and, if necessary, file a protest.

Clearly, no voter gets a copy of her own receipt to take home; she only gets copies of receipts from randomly chosen previous voters. That is, our "floating receipt" protocol satisfies both the *Anonymity* and *Exchange* properties. The protocol also satisfies the *Coverage* property, since, e.g, each of the first half of the deposited receipts has probability at least $1 - \prod_{m=(n/2)+1}^{n}(1 - 1/m) = 1/2$ of being selected at least once for copying. And it satisfies *Collusion-Resistance* since it is not clear from a receipt copy when the corresponding original receipt was deposited, and any voter since the original deposit might also have a copy of that receipt.

We note that even if *all* voters conspire to sell their votes, there is no way to prove how any one of them voted, even if they all reveal the receipt copies they brought home.

Floating receipts are quite powerful; in the next section we examine a voting system based almost entirely on the use of floating receipts.

5 "Twin"—a simple OneBallot voting protocol

Twin is a remarkably simple voting system based almost entirely on the power of floating receipts. It works for *any* voting system given the SBA. Basically, it simply employs the floating receipts scheme last section combined with *One*Ballot voting:

Each voter simply marks a ballot and puts it in a bin, receiving a copy of a random previously-cast ballot from that bin as her take-home receipt. (Ballots have ID numbers hidden under scratch-off. A checking machine can check that the marking is valid and the scratch-off remains unscratched, and only if so is the ballot placed in the bin; the scratch-off is automatically removed as it enters. The bin can be transparent enough so that everybody can see this all is happening, although not transparent enough to permit voters to read their ballot's ID number. Only voters later than the Tth receive take-home receipts, although an optional "phase III" can be added as in §4.1.)[13] The receipts can have notary-stamps or digital signatures added when they are given to the voter.[14] At the end of the day, all the ballots in the bin, and a list of all voter names and addresses, are posted on the PBB.

Twin is simple; the voter need not do any arithmetic, nor worry about strange consistency conditions for multiple ballots. You can't sell your vote because your receipt is a copy of somebody else's ballot. Talliers can't manipulate the PBB without risking detection, since there is a certified copy of most ballots somewhere and they don't know which ones and who has the copies. Talliers can't add or delete ballots (cf. §2.3.1). "Dumpster-diving" or other methods to collect discarded receipts, followed by cheating to alter the corresponding PBB ballots, are *unsafe* for cheaters because the collusion-resistance property of floating receipts.

One worry with Twin is that since voters cannot verify their *own* ballots, a small political party may feel that only a tiny fraction of its ballots are verified by loyal party members. But a major party may well be motivated to prevent fraud against a minor party in cases where it alters an election, and voter education should stress that voters should check whatever receipt they were given, as part of their civic duty.

6 Conclusion

Our new voting system, ThreeBallot, allows voters to verify that their votes are cast as intended, and to check that their vote is included in the final tally. All cast ballots are published, tampering with votes

[12] We could also add an optional **Phase III** in which the first T voters *return* to the polls at the end of the day and collect "delayed" receipts as in **IIa** – or we might only demand phase III for voting machines on which $< 2T$ voters voted (i.e. precisely those seen ex post facto to have violated our assumption that T was small compared to the number of voters).

[13] Juho Laatu and D.D.K.Sleator also emailed us some related proposals.

[14] Watchdog groups, or anybody with a computer, could provide publicly-usable digital-signature verifiers everywhere.

can be detected, and vote-privacy seems unbreachable, given SBA.

This is the first time such end-to-end verifiability has been obtained *without* the use of cryptographic techniques. Indeed the use of powerful computerized DRE machines with our protocols is actually *bad* because those computers might be, e.g, remembering things, such as ballot-triples, which would destroy our security. We require *simple* and *manifestly un-powerful* machines.[15] The principles employed by ThreeBallot are simple and easy to understand.

VAV increases ThreeBallot's applicability to handle *all* voting systems, and in some cases enlarges the set of elections for which the Short Ballot Assumption reasonably holds.

The notion of Floating Receipts is new here, and greatly strengthens the security properties of ThreeBallot and VAV.

Twin is a remarkably simple voting system, based almost entirely on Floating Receipts. It has practical potential.[16]

All three systems can be "mixed" with ordinary OneBallot voting as in §2.6.1; this exerts a protective effect on the ordinary ballots and provides an "easy upgrade path."

Acknowledgments

We are grateful for support from the National Science Foundation (NSF/ITR 0326277) and from the CalTech/MIT Voting Technology Project.

We thank Ben Adida, Andrew Appel, Jessy Baker, Alan Bawden, David Chaum, Ronald Crane, Chris Crutchfield, Kathy Dopp, John Kelsey, Jan Kok, Juho Laatu, Silvio Micali, Peter Neumann, Ben Riva, Alex Rivest, Michael A. Rouse, Julie Shamos, Michael Shamos, Emily Shen, Clay Shentrup, Daniel D.K. Sleator, Charlie Strauss, Jeroen van de Graaf, and Dan Wallach for feedback and comments. (But errors are ours.)

References

[1] Brennan Center Task Force on Voting System Security (L. Norden Chair) . The machinery of democracy: Protecting elections in an electronic world. Tech. rep., Brennan Center for Justice, 2006. http://www.brennancenter.org/programs/dem_vr_hava_machineryofdemocracy.html.

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

[3] Araújo, R., Custódio, R., and van de Graaf, J. A verifiable voting protocol based on farnel, June 2007. IAVoSS Workshop On Trustworthy Elections (WOTE2007), Ottawa, Canada.

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

[5] Aslam, J. A., Popa, R. A., and Rivest, R. L. Estimating the size and confidence of a statistical audit, 2007. To appear in Proceedings EVT07 (Boston, MA).

[6] Belote, G., Jones, H., and Juang, J. Threeballot analysis. Term paper presentation for MIT class 6.857 Fall 2006. http://theory.csail.mit.edu/classes/6.857/projects/threeBallotPresentation.pdf.

[7] Brams, S., and Fishburn, P. *Approval Voting*. Birkhauser, 1983.

[8] Chaum, D. Secret ballot receipts: True voter-verifiable elections. *IEEE J. Security and Privacy* (Jan/Feb 2004), 38 – 47.

[9] Chaum, D., Ryan, P. Y. A., and Schneider, S. A. A practical, voter-verifiable election scheme. Tech. Rep. CS-TR-880, University of Newcastle upon Tyne School of Computing Science, December 2004. http://www.cs.ncl.ac.uk/research/pubs/trs/papers/880.pdf.

[10] Cichoń, J., Kutylowski, M., and Węglorz, B. Anonymity of the ThreeBallot voting protocol, 2007. Available from authors.

[11] Devegili, A. J. 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.

[12] Gumbel, A. *Steal This Vote*. Nation Books, 2005.

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

[14] Jones, D. W. Chain voting, August 26, 2005. http://vote.nist.gov/threats/papers/ChainVoting.pdf .

[15] Jones, H., Juang, J., and Belote, G. Threeballot in the field, Fall 2006. Term paper for MIT course 6.857. http://theory.csail.mit.edu/classes/6.857/projects/threeBallotPaper.pdf.

[16] Karlof, C., Sastry, N., and Wagner, D. Cryptographic voting protocols: A system perspective. In *Proceedings 14th USENIX Security Symposium* (August 2005). http://www.cs.berkeley.edu/~nks/papers/cryptovoting-usenix05.pdf.

[17] Kiayias, A., and Yung, M. The vector-ballot e-voting approach. In *Proc. Financial Cryptography* (2004), Tsang and Wei, Eds., vol. LNCS 3110, Springer, pp. 72–89.

[18] Levin, J., and Nalebuff, B. An introduction to vote-counting schemes. *J. Economic Perspectives (special issue on voting systems) 9*, 1 (1995), 3–26.

[19] Naor, M., and Shamir, A. Visual cryptography. In *Proc. Eurocrypt '94* (1994), vol. LNCS 950, Springer, pp. 1–12.

[20] Nurmi, H. J. *Comparing Voting Systems*. Kluwer, 1987.

[15] However, *after* the ballots have been cast and checked and the voters have gotten their receipts, from *then onward* everything could be computerized, and it would not matter if those computers perhaps were penetrated, in the sense that anything they did wrong or fraudulently would be detectable.

[16] ThreeBallot, VAV, and Twin are hereby placed in the public domain—We are not filing for any patents on this approach, and we encourage others who work on extensions, improvements, or variations of this approach to act similarly. Our democracy is too important...

[21] Randell, B., and Ryan, P. Y. A. Voting technologies and trust. *IEEE Security and Privacy 4*, 5 (September/October 2006), 50–56.

[22] Rivest, R. L. The ThreeBallot voting system, October 1 2006. http://theory.csail.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf.

[23] Ryan, P. Y. A., and Peacock, T. Prêt à Voter: A system perspective. Tech. Rep. CS-TR-929, University of Newcastle upon Tyne School of Computing Science, September 2005. http://www.cs.ncl.ac.uk/research/pubs/trs/papers/929.pdf.

[24] Ryan, P. Y. A., and Schneider, S. A. Prêt à Voter with re-encryption mixes. Tech. Rep. CS-TR-956, University of Newcastle upon Tyne School of Computing Science, April 2006. http://www.cs.ncl.ac.uk/research/pubs/trs/papers/956.pdf.

[25] Saari, D. G. *Geometry of Voting*. Springer, 1994.

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

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

[28] Smith, W. D. Cryptographic election protocols for reweighted range voting & reweighted transferable vote voting. #90 at: http://www.math.temple.edu/~wds/homepage/works.html.

[29] Smith, W. D. New cryptographic voting scheme with best-known theoretical properties. Presented at FEE Milan 2005 and #89 at: http://www.math.temple.edu/~wds/homepage/works.html.

[30] Smith, W. D. Cryptography meets voting, September 10 2005. #80 at: http://www.math.temple.edu/~wds/homepage/works.html.

[31] Strauss, C. E. M. A critical review of the triple ballot voting system, part 2: Cracking the triple ballot encryption, Oct. 8 2006. Draft V1.5. http://www.cs.princeton.edu/~appel/voting/Strauss-ThreeBallotCritique2v1.5.pdf.