Punchscan with Independent Ballot Sheets: Simplifying Ballot Printing and Distribution with Independently Selected Ballot Halves

(extended abstract)

Richard T. Carback III[*], Stefan Popoveniuc[†], Alan T. Sherman[*‡], and David Chaum[§]
carback1@umbc.edu, poste@gwu.edu, dralansherman@starpower.net, david@chaum.com
Punchscan Voting Team – http://punchscan.org/

June 15, 2007

Abstract

We propose and implement a modification to the Punchscan protocol that simplifies ballot printing and distribution. In this improved version, each voter creates a ballot at the polling location by combining independently selected ballot halves, rather than using two pre-selected halves with the same serial number. The only time a ballot used for voting is human-readable is when it is in the voter's hands, reducing possible opportunities to violate voter privacy. This small but nontrivial change lets election officials print and distribute ballots using multiple printers more easily, without giving any one printer the ability to compromise voter privacy with certainty.

Keywords. Ballot printing and distribution, ballot privacy, election integrity, end-to-end voting (E2E), open-audit, PageScan, Punchscan with Independent Ballot Sheets (IBS), receipt-based voting, voting technology.

1 Introduction

In Punchscan [1, 6, 11], printers and anyone with access to printed ballots must be trusted not to violate voter privacy by recording the associations between ballots and the random permutations on them which determine how to interpret ballot marks. In this paper we explore the strategy of printing the top and bottom ballot halves separately, more easily allowing different printers to print each half.

Traditional Punchscan ballot halves could be printed separately, but doing so would require a complicated process of combining the separately printed ballot halves with identical IDs, and the people performing this process would have to be trusted.

Inspired by the concept of a binary weapon,[1] Sherman wondered if it would be possible to create a Punchscan ballot in the polling place by separately combining independently printed ballot halves, each with a separate ID, with the hope of reducing required trust in the printers and thereby enhancing ballot privacy. Sherman's hope was that alone, each ballot half would simply contain random permutations; only the combination of halves would require special privacy treatment.

We explore how to implement such a modification to Punchscan, which we call *Punchscan with Independent Ballot Sheets (IBS)*. We also analyze its benefits and costs, in comparison with traditional Punchscan.

[*] Cyber Defense Lab, Center for Information Security and Assurance., Dept. of Computer Science and Electrical Engineering. University of Maryland, Baltimore County (UMBC). http://cisa.umbc.edu/

[†] Dept. of Computer Science. George Washington University. http://www.cs.gwu.edu/

[‡] National Center for the Study of Elections, UMBC.

[§] Votegrity. http://votegrity.com

[1] In a binary chemical weapon, two chemicals are separately stored, each safe by itself. Only when combined do these two ingredients form a dangerous substance.

It turns out that the privacy properties of Punchscan IBS with separate printers and traditional Punchscan with separate printers are essentially the same, but that Punchscan IBS o□ers greater simplicity and flexibility in printing and distributing ballots.

In both systems, after marking a two-sheet ballot, the voter destroys one of the sheets. In Punchscan IBS, the ID of the destroyed sheet is copied onto the surviving sheet. For ballot privacy, in both systems it is important that an adversary cannot determine the random permutations on the destroyed sheet.

In the next section we briefly review related work. Next, we discuss what can be done to protect privacy with sound procedures. Then we summarize the Punchscan protocol and explain changes to support IBS. Finally, we discuss properties of the new system and state our conclusions.

2 Related Work

Invented by Chaum, Punchscan is an *end-to-end (E2E)* secure voting system in which each voter can verify that her ballot is correctly recorded, and observers can verify that all tallies are correctly computed. Using an optically-scanned paper ballot, it is more practical than Chaum's [2] original visual crypto voting system, and it improves upon Prêt à Voter (ready to vote) by Chaum, Ryan, and Schneider [3] by allowing candidates to be listed in a fixed order, as required by some states. In 2007, Punchscan was successfully used in student elections at the University of Ottawa [1]. Other E2E systems include VoteHere by Andrew Ne□ [9, 10], and ThreeBallot, VAV, and Twin by Rivest and Smith [15, 12] which aim to avoid mathematical cryptography.

Like Punchscan, Prêt à Voter has a nonuniform ballot (every ballot is di□erent) that includes random permutations, and has similar printing issues. Ryan [13, 14] discusses ballot printing and distribution issues for Prêt à Voter, but the separation of Punchscan sheets o□ers a unique protection and auditing mechanism. ThreeBallot uses three identical ballot sheets attached together. While it does not use random permutations, it also has similar printing issues, as prior knowledge of what Ballot IDs are attached together can violate voter privacy. However, this vulnerability might be ameliorated with a similar independent ballot selection strategy.

Fisher [5] was the first to explore changing the Punchscan system to support combining top and bottom sheets with independently chosen serial numbers. His proposal, PageScan, did not scale as well as does our solution. To count ballots verifiably, Punchscan uses three tables of secure bit commitments [8]. Many versions of one of the tables are created, and to do so requires two column-wise commitments for each iteration. PageScan changed this by doubling the number of rows in that table and requiring two cell-wise commitments for each row of that table. Our solution doubles the number of rows, but maintains two column-wise commitments per iteration.

Our work also touches on the following two themes, which are well known in voting research folkore. First, in any voting system, printers (and more generally, all output devices) are potential threats to ballot privacy. The details depend on the system and location of the printer. Whereas optical scan ballot markers and VoteHere use printers in the polling place, Punchscan and Prêt à Voter use printers to print ballots that are delivered to the polling place.

Second, whenever the voting process permits random choices to be made, there is a potential threat for a subliminal channel that leaks private data through the random choices (*e.g.*, see Karlof *et al.* [7]). Ryan and Peacock [14] showed that systems with nonuniform printed ballots help avoid such attacks because the random information is printed on the ballots before being given to the voter. To prevent subliminal channels, computers and humans must not be allowed to choose their own random bits; random bits should be determined before voting and later be verified through a secure bit commitment scheme. This principle is why the Punchscan protocol forces the voter to commit to which sheet to destroy before marking the ballot, as enforced by a clipboard with physical lock. Following this principle, in Punchscan IBS, no voter should be allowed to choose entirely by herself

which ballot halves to use.

3 Procedurally Protecting Voter Privacy

Procedural privacy protections follow two main strategies: disassociation and distribution. We now discuss each of these strategies and how they might be implemented in Punchscan.

3.1 Disassociation

Disassociation is any strategy that conceals voter identity from ballot serial numbers on the receipts, making it harder to determine what ballot a voter used without observing her in the voting booth. It can be implemented in three ways with varying properties. The least complex way is to protect the association made between the voter and the ballot serial number, and can be accomplished by preventing poll workers from recording such data. In Punchscan, overcoming this protection would require that an attacker physically find the voter with her ballot and determine the contents of the destroyed sheet.

Second, the voter could exchange her receipt with that of another voter. Each of them must trust that the person they exchange with will check the receipt and not tell (or be able to tell) the attacker the identity of the other voter. For example, the polling place could have a box or some mechanism that would exchange the current receipt with a previous one at random.

Third, instead of leaving with a receipt, each voter could be made to give it to a third party whom they trust before leaving the polling place [4]. The third party would then be responsible for checking voter ballots and protecting voters from attackers.

3.2 Distribution

Distribution works by delegating the printing and storing process among multiple trusted parties. Attackers must then compromise multiple sites, increasing the amount of resources an attacker must use to be e□ective. This strategy can be implemented in Punchscan by distributing the printing and storage process among multiple trusted printers, and by having each printer print only one sheet of the two-sheet ballot.

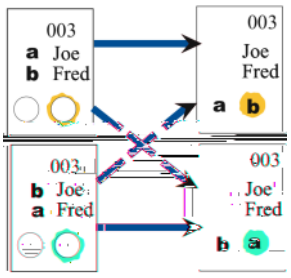

Figure 1: **The Punchscan Ballot.** Whether the vote was for Joe (solid arrows) or Fred (dashed arrows) can only be determined when both sheets of the ballot are available.

Distributing the printing process limits the effectiveness of an attack to those ballots printed by each compromised printer or held by a compromised storage facility. Printing each sheet separately requires at least two sites to be compromised to guarantee success in violating voter privacy: with two printers, for each voter there is fifty percent chance that the printer will have printed the discarded sheet and thus be able to read the voter's marked ballot from her receipt. Admittedly, a fifty percent reduction in the chance that a corrupt printer could violate a voter's privacy is not an impressive improvement.

4 The Punchscan Protocol

We now briefly review the Punchscan ballot and explain the Punchboard, the structure used to count the ballots verifiably. Subsequently, we discuss the di□erent audits that provide integrity and privacy in the system. For more in-depth explanations, see [6, 11, 1]. Readers who are already familiar with Punchscan are urged to skip this section.

4.1 Ballot

The Punchscan ballot is created by combining a top and bottom sheet of paper. The top sheet has letters next to candidate names and holes in it to show letters that are printed on the bottom sheet. The letters on both sheets are ordered randomly.

To vote, each voter uses a bingo dauber to mark the letter on the bottom sheet that is next to the candidate of her choice on the top sheet.

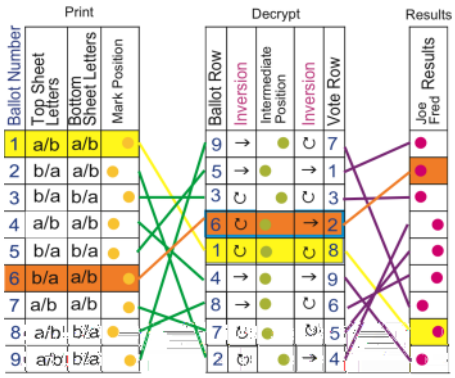

Figure 2: **The Punchboard**. This structure permits the election authority to determine how voters voted based on the position marked by the voter on her receipt (left or right in this example). The contents of this structure are initially concealed and partially revealed in an auditing process that protects voter privacy but ensures used ballots were correctly tabulated.

Afterwards, either the top or the bottom sheet is destroyed, and the surviving sheet is scanned, publicly posted, and kept by the voter as a receipt. As shown in Figure 1, neither half of the ballot can reveal the original vote by itself. Only the *Election Authority (EA)* can determine the original intent, and it does so using the Punchboard. The position marked by the voter is known as the mark position, and in subsequent diagrams is either "left" or "right," but for races with more than two candidates, is generally considered a 0 or 1 indexed array starting with the first number at the leftmost position.

4.2 Punchboard

In order to determine voter intent, election officials must know the letter ordering on the destroyed half of the ballot, and this information is available through the Punchboard, shown in Figure 2. To interpret results from this, candidate order is associated with a marked position in the Results table. Thus, a dot in the left position in the Results table represents a vote for the first candidate listed on the ballot (Joe).

The Punchboard is used to provide voter privacy and election integrity. If we post it as shown in the figure, there is no privacy in the system, but if it remains secret, we provide no publicly verifiable integrity to the counting process. In order to achieve both of these properties, Punchscan uses its own unconditionally secure bit commitment scheme to commit to certain data before ballots are printed for the election. This method enforces integrity by making public certain values as we progress through the election, allowing anyone interested to check to make sure the public values, or revealed data, match what election authorities committed to before the election. The data not made public protect the privacy of voters. The initial Punchboard commits each top and bottom cell in the print table, each row of all three tables, and the two columns on each side of the Intermediate Position column in the Decrypt table. Multiple versions of the Decrypt table are published.

4.3 Auditing

There are three types of audits: pre-election, results posting, and post-election. Because a malicious person does not know what data will be chosen by the auditors, any malicious action taken has a high risk of being caught. Thus, it is important that the EA commit before auditors perform any actions, because prior knowledge of intended auditor actions would let the EA or the attackers know what malicious actions they could take without being detected.

Pre-Election Audit. The pre-election audit ensures proper construction of the Punchboard. Auditors choose half of the rows at random and the EA publishes the contents of those rows. The published rows are checked with their commitments to ensure that they are well-formed, they are then discarded and the remaining rows are used to print the sheets that make up each ballot used in the election. This audit makes half of the rows unusable, so the EA must generate at least twice as many rows as the number of needed ballots. Although the EA may prefer to perform this audit before the election, all of this checking could be performed as part of the post-election audit.

Posting Results. When results are posted, election o□cials populate the Mark Position, Intermediate Position, and Results columns of the tables. They additionally reveal the sheet that

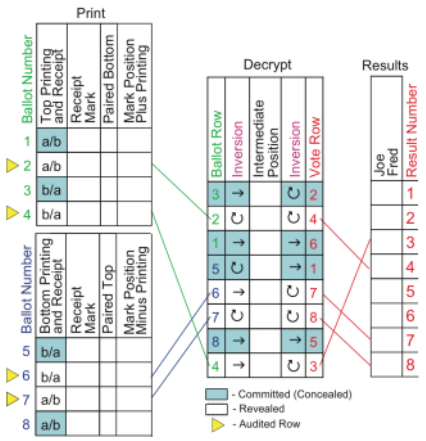

Figure 3: **Pre-Election**. The Punchboard after the pre-election audit. The data in half of the rows are posted so the public can verify that the Punchboard is well-formed.

each voter took home as a receipt. Each voter is able to verify that her ballot was included with the correct marks in the final tally, that her receipt matches the revealed data, and that it was well-formed. Everyone is able to verify that revealed data matches what was committed to before the election.

Post-Election Audit. The post-election audit ensures that the counting process was executed properly. For each published Decrypt table, auditors choose the two columns left or right of the Intermediate Position column and the EA reveals that data. That way, everyone can then check that the marked positions match the intermediary values, or that the intermediary values match the final results. Because the EA or attackers did not know what half of each Decrypt table will be selected before they populate the Intermediate Position and Results columns, improperly publishing a result in either column would result in an overwhelming probability of being caught.

5 The Independent Ballot Sheet Protocol

We now present our proposed modification and show that our method allows us to maintain auditing and integrity properties that are at least as strong as those in traditional Punchscan. Our modification does not change the way people vote, but it does require the sheets to be combined in the polling booth and that the serial number of the destroyed sheet be recorded onto the receipt. It also changes the way the Punchboard is structured and used, and the meaning of its tables.

In the original system, both the Print and Decrypt tables had combined sheets represented in each row, but now each row represents a single half-sheet. The structure of the Punchboard Decrypt (D) and Results (R) tables remain the same, but the Print (P) table changes and the number of rows in all of the tables are doubled. The new P table has 4 columns. The first column, $P1$, records letter order on either a top or bottom sheet. $P2$ records the position marked by the voter after if that sheet is taken by the voter as a receipt. $P3$ records the sheet that the current sheet was paired with when it was used. $P4$ records the mark position after the value in the receipt, $P1$, is removed from the recorded mark position, $P2$.

To generate the Punchboard, let n be the number of ballots that will be available for voters. The election authority (EA) then generates $2n$ virtual top pages and $2n$ virtual bottom pages and puts them in the P table. For simplicity, we will assume the top pages are in the first rows of the P table (positions 1 to $2n$, therefore having serial numbers from 1 to $2n$) and the bottom pages are in last rows (positions $2n$ + 1 to $4n$, therefore having serial numbers from $2n$ + 1 to $4n$). The EA creates a D table where each row will correspond to a row in P. Therefore half of the rows in D will correspond to top pages and the other half to bottom pages. The EA commits to the rows that this creates, just as in the previous protocol. The rows in D are then shu□ed and the commitments to the rows are published.

Pre-Election Audit. Figure 3 illustrates the pre-election aduit. In the pre-election audit, the auditors choose n top sheets and n bottom sheets from the P table. The election authority opens the rows in P and the corresponding rows in D. Anyone can check the commitments and the fact that $P1 = D2 \oplus D4$ ($\oplus$ meaning the commutative composition operation), *i.e.* that the value to be printed matches the sum of the two inver-

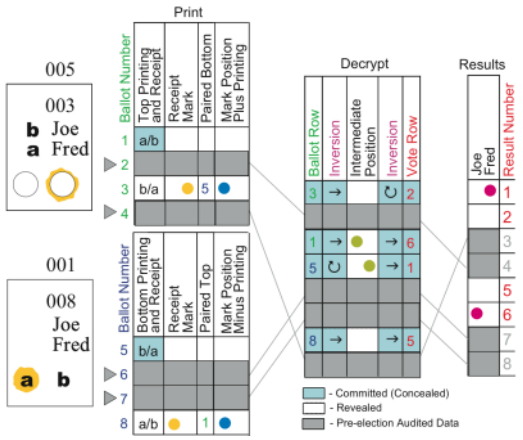

Figure 4: **Results.** The Punchboard after results are posted. Half of the Mark, Intermediate, and Results columns are populated to give unaudited results of the election. Note that sheet 003 was paired with sheet 005, and the number 5 appears in the paired top column. Likewise, sheet 008 was paired with top sheet 001, and it appears in 8's paired sheet column.

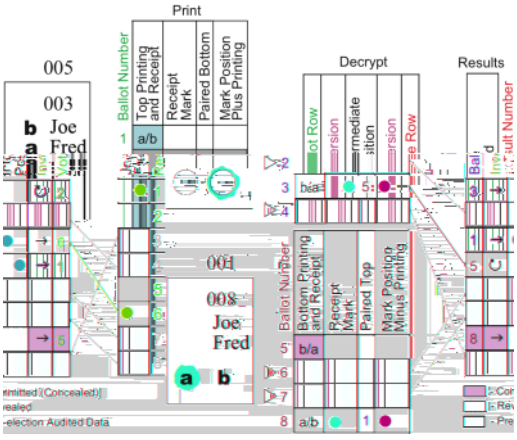

Figure 5: **Post-Election Audit.** The Punchboard after the post-election audit. Data to the left or right of the Intermediate Position of the Decrypt table are revealed to audit the results of the election.

sions. This slightly altered process produces the same result as before, with half of the possible number of ballots being discarded to verify that the Punchboard is well-formed.

Posting Results. At this point, the ballot pages are printed and any top page can be combined with any bottom page. The voting procedure is the same as before. In addition to the voter marks and the serial number, the receipt must also contain the serial number of the sheet it was paired with that has been destroyed. This could be done by not destroying the serial number of the destroyed sheet, or by copying the serial number to the receipt and signing it for authenticity.

Now we have a correspondence between the receipt $P1$ and the sheet it was paired with, $P3$. The receipt the voter took home is revealed by $P1$, and the discarded sheet serial number is $P3$. $P4$ represents the ballot only taking the destroyed sheet into account. That is, $P4 = P2 \oplus P1$ for each row in P with a populated $P2$. For example, if the mark is left and the receipt is an inverting page, the $P4$ column contains a right mark. Once the receipts are published, anyone can compute $P4$. This is illustrated in Figure 4. Also during this time, the EA computes $D3$ and R, filling in the Intermediary and Results values. Note that when counting, $D1$ now points to the value in $P3$, not $P1$ as in the pre-election audit.

As before, each voter is able to verify her ballot is included in the tally and is well-formed, and everyone is able to verify that $P4$ was computed correctly and that the revealed data match commitments.

Post-Election Audit. Figure 5 illustrates the post-election audit. This figure shows selected row halves of either the two columns left or right of $D3$ being posted. In the actual method, where multiple D tables are published, the entire columns to the left or right are posted. The post-election audit remains virtually unchanged from the original scheme. The major difference is that the audits reveal information that is not necessarily needed to verify integrity of the system because that information is revealed in the results phase when receipt values are posted. This information is denoted by the blank mark in the intermediate and results cells in the tables.

6 Properties of Independent Ballot Sheets

Our proposed modification doubles the number of rows in every table of the original system, and adds two columns. The result is that the the ballot is only "human-readable" when it is in

the voter's hands, and the auditing and privacy properties of the original scheme are maintained.

6.1 Usability

The ballot sheets must be combined, and the serial number of the destroyed sheet must be copied over to the receipt. By contrast, with traditional Punchscan, the ballot sheets are already joined and each sheet has the same serial number. It is a slight improvement not to require voters or poll workers to ensure that serial numbers are identical, but the other two tasks cause additional complexity. We believe that there is no reason it cannot be done in a mechanically robust way. Our suggestion is to utilize the help of poll workers, special packaging, and a clipboard similar to that used in traditional Punchscan.

The clipboard allows poll workers to set the ballots properly, and the special packaging conceals the ballot contents. The serial numbers can be printed on removable stickers. The poll worker would ask the voter which sheet she would like to destroy, pull the serial number from that sheet and attach it to the sheet used for the receipt, and attach both to the clipboard. Once in the polling booth, each voter would pull o the packaging, vote, and destroy their half of the ballot as in traditional Punchscan.

6.2 Advantages in Distribution Procedures

A key innovation of our protocol is that it permits great flexibility in ballot distribution strategies. Election o cials may easily arrange for different printers to print various top or bottom sheets. The more printers used, the less likely an attacker will be able to conduct a targeted attack because he will not be able to ensure that the target voter will receive and choose to destroy a ballot sheet that they have compromised.

The system also benefits from increased reliability. Whereas the old method requires printers to deliver the correct ballot sheets to the same polling locations, it does not matter where polling sheets come from in Punchscan IBS.

6.3 Privacy

Our modification does not improve voter privacy over the original system without the implementation of distribution strategies. To see the differences, consider the following three cases:

1. Punchscan with 1 printer.
2. Punchscan with 2 printers.
3. The proposed system with 2 printers.

In Case 1, ballot sheets are created, printed, and stored together. If any of these sheets are compromised, an attacker knows the information on both sheets and can determine the meaning of marks on any receipt. Cases 2 and 3 o er a distinct advantage. By separately printing the top and bottom sheets, an attacker gains access only to either all the top or all the bottom sheets. Thus, if the voter takes home an uncompromised sheet, the attack does not succeed. On the other hand, a coercion attack might still work if the victim is unwilling to risk that the coercer has compromised the correct sheet. Case 3 provides extra flexibility over 2 yielding a marginal advantage, because there is no need to ensure that matching ballot serial numbers are combined.

6.4 Printing

Punchscan IBS prints on only one sheet of paper at a time, creating some challenges and benefits. Using di erent printers increases the chance of printing errors that produce unusable ballot matches. On the bottom sheet, too much skew can misalign letters from their corresponding top sheet holes. Also, the need to package each sheet securely increases cost.

Benefits include the following. Cost is marginally reduced by not having to fold the sheets. Feeding one sheet instead of two into the printer is generally more reliable. This method also lessens the severity of privacy leakage when information from one sheet may inadvertently transfer to the other sheet during the printing process. Finally, distributed printing processes are more resistant to disruption.

7 Conclusions and Open Problems

Our modification, Punchscan IBS, enables election o cials easily to print the top and bottom

sheets separately, complicating attacks on ballot privacy. By contrast, such a printing strategy is not possible with Prêt à Voter.

Other variations to Punchscan might also be worth investigating, including printing ballots at each polling place and using a three- or four-sheet ballot. Printing ballots in advance, however, increases reliability and permits voters to daub their ballots even if all electronic equipment fails on election day. A three-sheet ballot would enable even greater distribution of printer trust but complicate a system already considered by some to be moderately complex.

Punchscan IBS exploits Punchscan's two-sheet ballot to permit distributing trust among multiple printers more easily than in traditional Punchscan. More field testing is required to gauge how well voters and election o□cials will handle this high-integrity voting system.

8 Acknowledgments

We thank Russ Fink, Aleks Essex, Jeremy Clark, John Krautheim, and the referees for helpful comments.

References

[1] *Punchscan Website*, `http://www.punchscan.org/`, November 2006.

[2] David Chaum, *Secret-ballot receipts: True voter-verifiable elections*, IEEE Security and Privacy **02** (2004), no. 1, 38–47.

[3] David Chaum, Peter Y.A. Ryan, and Steve A. Schneider, *A Practical, Voter-verifiable, Election Scheme*, Technical Report Series CS-TR-880, University of Newcastle Upon Tyne, School of Computer Science, December 2004.

[4] Lillie Coney, *Personal communication*, Electronic Privacy Information Center, National Committee for Voting Integrity, August 2006.

[5] Kevin Fisher, *Punchscan: Security Analysis of a High Integrity Voting System*, Master's thesis, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, December 2006.

[6] Kevin Fisher, Richard T. Carback III, and Alan T. Sherman, *Punchscan: Introduction and System Definition of a High-Integrity Election System*, Preproceedings of the 2006 IAVoSS Workshop on Trustworthy Elections (Robinson College, Cambridge, United Kingdom), International Association for Voting System Sciences, 2006.

[7] Chris Karlof, Naveen Sastry, and David Wagner, *Cryptographic Voting Protocols: A Systems Perspective*, Fourteenth USENIX Security Symposium, August 2005.

[8] Adrian Kent, *Unconditionally Secure Bit Commitment*, Physical Review Letters **83** (1999), no. 7, 1447–1450.

[9] C. Andrew Ne□, *Practical high certainty intent verification for encrypted votes*, `http://www.votehere.net/vhti/documentation`, October 2004.

[10] _____, *Verifiable mixing (shuffling) of ElGamal pairs*, `http://www.votehere.net/vhti/documentation`, April 2004.

[11] Stefan Popoveniuc and Ben Hosp, *An Introduction to Punchscan*, Preproceedings of the 2006 IAVoSS Workshop on Trustworthy Elections (Robinson College, Cambridge, United Kingdom), International Association for Voting System Sciences, 2006.

[12] Ronald L. Rivest, *The Three-Ballot Voting System*, `http://theory.csail.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf`, October 2006.

[13] Peter Y. A. Ryan, *Prêt à Voter with Pallier Encryption*, Technical Report Series CS-TR-965, University of Newcastle Upon Tyne, School of Computer Science, June 2006.

[14] Peter Y.A. Ryan and Thea Peacock, *Prêt à Voter: A Systems Perspective*, `http://www.cs.ncl.ac.uk/research/pubs/trs/papers/929.pdf`, September 2005.

[15] Ronald L. Rivest & Warren D. Smith, *Three Voting Protocols: Threeballot, VAV, and Twin*, Usenix/Accurate EVT, August 2007.