Prêt à Voter with Write-Ins

Steve Schneider, Sriramkrishnan Srinivasan, Chris Culnane, James Heather, and Zhe Xia

University of Surrey, Guildford, Surrey, GU2 7XH, UK

Abstract. This paper presents an extension of the Prêt à Voter verifiable voting system to handle write-ins. This is achieved by introducing an additional 'Write-In' option and allowing the voter optionally to enter a write-in candidate of their choice. The voter obtains a receipt which includes their write-in, but that receipt does not indicate whether the write-in candidate was selected or not. The system provides flexibility with respect to the tallying of write-in votes. We also introduce null ballots in order to achieve receipt-freeness with respect to write-ins.

Keywords: end-to-end verifiable voting, Prêt à Voter, write-in votes, receipt-freeness

1 Introduction

End-to-end verifiable voting systems provide mechanisms for verifying elections. These generally involve checking cast votes against some evidence provided to the voter, such as a receipt of voting, or a code number. 'Receipt-freeness' is an important property of such systems. Receipt-freeness requires that the voter's evidence, together with the evidence of integrity of the election provided by the authorities, should not give away how the voter voted. This protects against vote selling and voter coercion.

Elections allowing write-in candidates pose a particular challenge for end-to-end verifiable voting systems. A voter requires evidence of how she voted, to verify the vote was counted correctly and to be able to challenge the election if it was not. In the case of a write-in candidate, this means that the name entered may well be part of the voter's retained evidence. The election authorities may also need to reveal the name in order to demonstrate that the write-in votes were correctly tallied. Thus such a system can allow a voter to enter some information of her choice into her vote, in a verifiable way. The challenge is to do this so as to minimise the leakage of information. In particular, as noted in [8], it is not acceptable for a receipt to provide evidence of having voted for a particular write-in candidate.

Elections allowing write-ins are rare outside the U.S., but they are common in the U.S. as part of the electoral landscape. Their implementation in the U.S. varies widely[11], since rules regarding write-ins are determined at state level. For example, in the 2008 Presidential election, some states did not allow write-in candidates; in others, write-in candidates had to register before the election; and in others, any write-in is considered a legitimate candidate (subject to being eligible for office). It appears common practice that the totals for legitimate write-in candidates are reported as part of the election result. However, reporting of results for 'unofficial' write-in candidates may or may not occur. Write-ins are typically handled by a variety of technologies, including paper ballots and electronic voting systems.

There are therefore a range of possible legal requirements with respect to the results that must be reported. If a verifiable voting system is to support write-ins, then ideally it will not tally or report on more results than are required by the electoral rules. This will prevent unnecessary information leakage, as well as avoiding any unnecessary processing of votes. Hence it is desirable for such a system to provide flexibility with respect to the processing of the write-in votes.

The approach proposed in this paper also makes use of a 'write-in' option as an additional choice on a Prêt à Voter ballot form. The scheme we propose includes the write-in on the receipt and also on the bulletin board. This enables changes to any write-in image to be challenged by the receipt-holder. The inclusion of the write-in on the voter's receipt introduces additional challenges that we address by incorporating the write-in into the mix. Election officials allocate write-in votes to candidates, and they are tallied homomorphically to protect voters' privacy. Even if the receipt contains a recognisable write-in image which can be linked to a specific voter, the tallying procedure does not provide evidence of whether that vote was cast for that write-in candidate. We also introduce dummy receipts to allow a voter to vote for one write-in and obtain a receipt for another. This provides a stronger form of receipt-freeness with respect to write-ins than previous schemes.

Write-ins were first considered explicitly for cryptographic secure ballots in the Vector-Ballot scheme of [9], which uses a combination of mix networks and homomorphic encryption. It is noted in [2], that write-ins can potentially be incorporated into schemes such as Scratch&Vote [2], Prêt à Voter [16] and Punchscan [5], by including a 'write-in' option alongside the listed candidates, and incorporating a process such as the Vector-Ballot approach for handling the votes that are written in. This approach was required for the Takoma Park election [4] which was run using Scantegrity II [6]: in that election, voters ranked the candidates in order of preference, and so the system was extended to permit a write-in candidate in any position. This was achieved by incorporating a 'write-in' option and allowing it to be selected for any rank in the list. The associated write-in entry was scanned into the system, and processed by an 'Election Resolution Manager' component in order for it to be incorporated into the count. We discuss these schemes and compare them to ours in Section 5.

This paper is structured as follows: Section 2 describes the ballot form and how votes are cast, and discusses the nature of the receipt and the information it contains; Section 3 describes how the votes are processed through the two mixnets; Section 4 discusses the resulting scheme with respect to receipt-freeness; finally, Section 5 discusses other aspects of the scheme, and provides a comparison with other verifiable election schemes.

2 Incorporating write-ins into Prêt à Voter

Prêt à Voter uses cryptography in the construction of ballot forms, and in the processing of votes. The key features we require of the cryptosystem for the scheme presented in this paper are:

- probabilistic and semantically secure: that the same plaintext can be encrypted in many different ways, using different randomisation values, and that it should be computationally infeasible to tell whether two ciphertexts correspond to the same plaintext, without knowledge of the secret keys or randomisation involved;
- that a ciphertext can be re-encrypted without knowledge of the private key;
- that the cryptosystem is additively homomorphic: ciphertexts containing integers can be combined to a ciphertext containing their sum, without knowledge of the private key.

These properties are provided by the Paillier cryptosystem [**?**], and also by the ElGamal cryptosystem[**?**] using exponents to encode integers (since ElGamal itself is multiplicatively homomorphic).

In this paper we use the notation $E(m, r)$ to denote a message m encrypted under the election key with randomisation r. Re-encryption is denoted by changing the randomisation value, thus $E(m, r')$ is a re-encryption of $E(m, r)$ when r' is derived from r, and they both encrypt m. Decryption requires a threshold set of trusted parties, as discussed in [16].

2.1 The ballot form

The scheme incorporates write-ins by making an adaptation to the ballot form. Following the approach taken in [20], the ballot form is constructed with a randomised list of choices on the left hand side (and serial number at the top right) as illustrated in Figure 1. For plurality voting the possible permutations of the candidates could be simpler, as discussed in [18], but for this paper we will retain arbitrary candidate lists. The information capturing the candidate list is embedded in encrypted form in a code, illustrated here on the bottom of the ballot form. For each position on the ballot this gives an encryption of the associated candidate.

In common with other approaches, we introduce an additional 'candidate', 'Write-In', to the list of candidates on the left-hand side of the ballot, and include a space for a write-in candidate's name on the right-hand side of the ballot. Recall that the left-hand side is to be destroyed, and the right hand side will be scanned into the system and recorded as the vote. Completing the ballot form is carried out as in previous versions of Prêt à Voter[16], with the added opportunity to write-in a candidate: the voter finds their choice on the randomised list of candidates (which now includes the additional choice 'Write-In') and marks an 'X' in the corresponding box on the right-hand side; the voter may also write any name into the write-in box, which is on the right hand side.

2.2 Completing the ballot form

Figure 2 illustrates four ways of completing a valid ballot form. All except (d) are valid votes, though (c) also includes a decoy name (i.e. a write-in name who does not receive the vote).

- In vote (a), the voter has voted for Cathy, by selecting the box against Cathy's name. No name has been written in the write-in box.

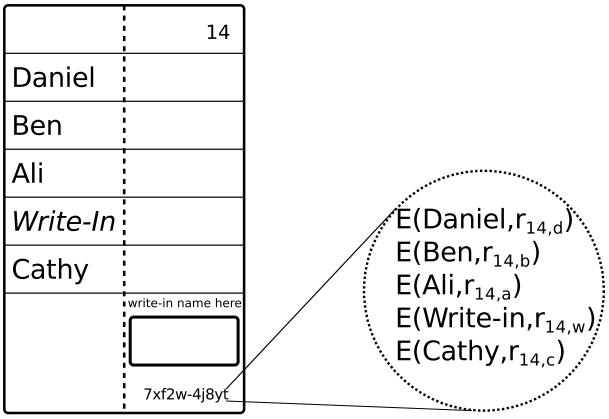

Fig. 1. Prêt à Voter ballot form with write-ins

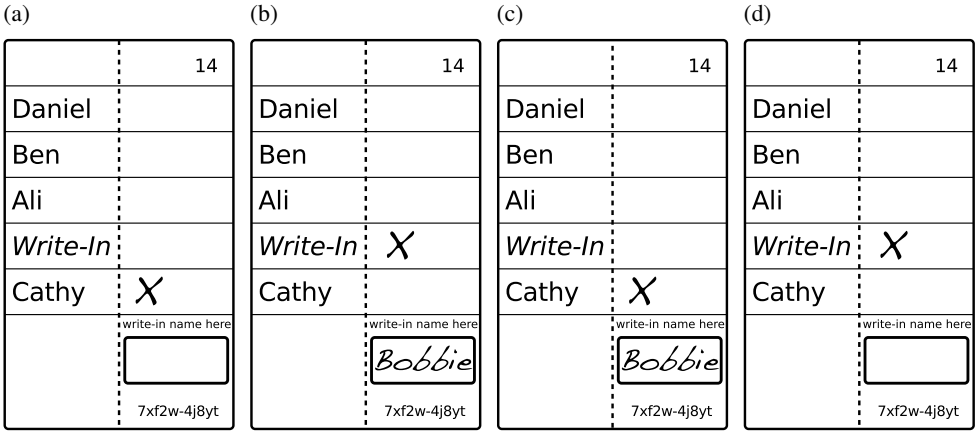

Fig. 2. Four possible ways of completing the ballot form

- In vote (b), the voter has chosen to cast a vote for a write-in candidate Bobbie. To do this, the voter selects the box against the candidate 'Write-In', and writes the name 'Bobbie' into the write-in box on the right-hand side of the ballot form,
- In vote (c), the voter has voted for Cathy. The name 'Bobbie' in the write-in box will not be counted as a vote for Bobbie, because the Write-In option was not selected.
- In vote (d), this is effectively a spoiled ballot: the write-in box was selected, but no name has been provided.

2.3 Dummy ballot forms

The scheme introduces the idea of dummy ballot forms, used to cast null votes. Here we first present the full scheme in which dummy ballots are provided to each voter. In Section 4.2 we discuss the alternative approach of making them optional for voters, and the consequences of not including them at all. Their purpose is to allow voters to obtain a receipt of any form with any write-in, while voting in any way they choose. A dummy ballot form is pictured in Figure 3. Following the construction of valid ballots, the code at the bottom of the form associates a (different) encryption of 'null' with each position.

2.4 Casting a vote

To cast their vote, voters are given a valid ballot form and also a dummy ballot form. There should be no relationship between the serial numbers of the two forms. They complete both, detach and destroy both left hand sides, scan both right hand sides into the system, and choose one receipt to retain. The receipt can be either that from

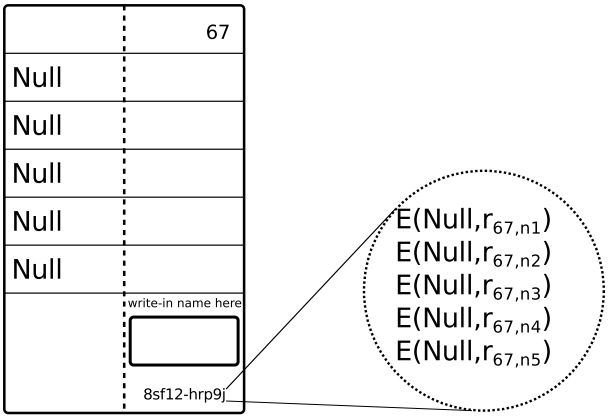

Fig. 3. Prêt à Voter dummy ballot form

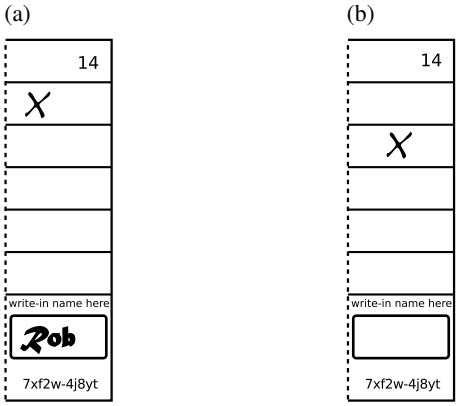

Fig. 4. Two possible forms of the right-hand side

the valid ballot form or from the dummy ballot form. The system must not know which receipt has been retained. One way to achieve this is to produce both receipts, and then require the voter to destroy one.

The right-hand side will either contain a write-in name, or not, as pictured in Figure 4.

- If the right-hand side contains a write-in name, as pictured in Figure 4 (a), then either it is a vote for that candidate, or it is a vote for one of the listed candidates as illustrated in Figure 2, or else it is a null vote from a dummy ballot form. Without decrypting the code, and in the absence of the left-hand side, there is no way to determine which selection was made. Hence this right-hand side does not provide evidence that the written-in candidate actually received the vote.
- If the right hand side does not contain a written-in candidate, as pictured in Figure 4 (b), then this vote is either a vote for one of the listed candidates, or a spoiled ballot, as illustrated in Figure 2, or else a null vote from a dummy ballot form. It does not indicate which of these possibilities actually holds.

3 Processing the votes

Following recent versions of Prêt à Voter [16, 17, 15], we use a *re-encryption* mixnet [14, 19, 12] to process the votes.

3.1 Anonymising the votes

A key feature of Prêt à Voter is the use of a public bulletin board to display the votes that are cast—the marked-up right hand sides of the ballot forms that the voters have submitted to the system. In this section we will consider

that the record of any write-in will be recorded on the bulletin board as an image of the write-in box. The scheme works the same way if the write-ins are in text form.

When a vote v_i is scanned into the system, it is written to the bulletin board against the serial number on the ballot form, and the voter retains a (signed) receipt, which matches the bulletin board entry. What is recorded on the bulletin board and on the receipt is the onion associated with the voter's selection, together with the readable image I_i of the entry in the write-in box. The onion for selection j on ballot form i is $E(C_j, r_{i,j})$: the selected candidate C_j in encrypted form with randomisation $r_{i,j}$.

At the end of polling, the votes are prepared for mixing by encrypting each serial number i with fixed randomiser 1, to obtain $E(i, 1)$. The results of this step are posted on the bulletin board and are easily verifiable. The entries to the mix are thus pairs of the form $(E(C_j, r_{i,j}), E(i, 1))$.

In principle the scheme could pass the scans I_i through the mix rather than pass the serial number i. However, a scan image is likely to be a bitmap of several kilobytes, and there will be practical problems encrypting such a large item and passing it through the mix. Using a serial number of a few tens of bits is significantly more efficient and less cumbersome.

The serial numbers are re-encrypted at each stage alongside the choice of candidate. At each stage of the mix each vote consists of an encrypted candidate and its associated encrypted serial number. The mix must not split the pairs at any stage. This can be achieved using the protocol of [12, **?**], or can be audited using mixnet techniques such as randomised partial checking [7]. Finally, the shuffled and re-encrypted votes are output from the mix, as another list of pairs of the form $E(C_j, r'_i), E(i, s'_i)$.

Tallying the listed candidates

At this stage, the encrypted candidates $E(C_j, r'_i)$ are decrypted in the usual way (by a threshold set of parties) to extract the candidates C_j that were selected. The serial numbers $E(i, s'_i)$ are not yet decrypted. The resulting list $\{(C_j, E(i, s'_i))\}$ is published. This process is illustrated in Figure 5.

After this stage, the votes have been decrypted, and so the election can be tallied. Any vote C_i for a listed candidate can be counted as a vote for that candidate. Any null vote is not counted against any candidate.

Tallying the write-ins

Any vote C_i for 'Write-In' cannot yet be counted, since the serial number linking to the image is encrypted and not yet available. It remains to tally the write-in candidates.

To obtain the write-in votes, we make use of a second mixnet. We use a homomorphic encryption scheme such as exponential ElGamal to encrypt the values 0 and 1 with a fixed randomiser, 1, to obtain known encryptions $E(0, 1)$ and $E(1, 1)$. This will enable us to obtain the aggregate values for each candidate. We take the list of results with their encrypted images $\{(C_j, E(i, s'_i))\}$ to generate a new list $\{(E(b_j, 1), E(i, s'_i))\}$, where each $E(b_j, 1)$ is the known encryption of 0 or 1, chosen as follows:

- if C_j is one of the listed candidates, then include $(E(0, 1), E(i, s'_i))$ in the input for the second mixnet;
- if C_j is 'Null', then include $(E(0, 1), E(i, s'_i))$ in the input for the second mixnet;
- if C_j is 'Write-In', then include $(E(1, 1), E(i, s'_i))$ in the input for the second mixnet

We include all the votes as input to the second mixnet: it is important to treat the decoy write-ins in the same way as the genuine write-in votes, otherwise the decoys will be exposed as such (because i would not eventually be revealed).

The resulting list is run through the second mixnet, and is re-encrypted and shuffled. The resulting elements are of the form $(E(b_i, t_i), E(i, s''_i))$. The $E(i, s''_i)$ are now decrypted, to obtain all of the write-in entries. Note that the encrypted 0 or 1 values indicate whether or not those write-in entries corresponded to a 'Write-In' selection, i.e. whether that entry should be counted in the tally. This process is illustrated in Figure 6.

The write-in entries are examined, and assigned to particular candidates. This is the stage where election officials will need to make judgements as to the 'voter's intention' in each case, and where procedural and legal challenges might take place. In due course, all of the write-ins are allocated to particular candidates. This process is independent of the processing of the ballots through the mixes, and can be carried out concurrently.

The encrypted 0 or 1 values for each write-in candidate to be tallied can then be combined homomorphically: adding them together and then decrypting the result. This yields the total number of votes selected as 'Write-In' for that candidate, but it does not reveal for any particular vote whether that was a valid write-in, or a vote for one

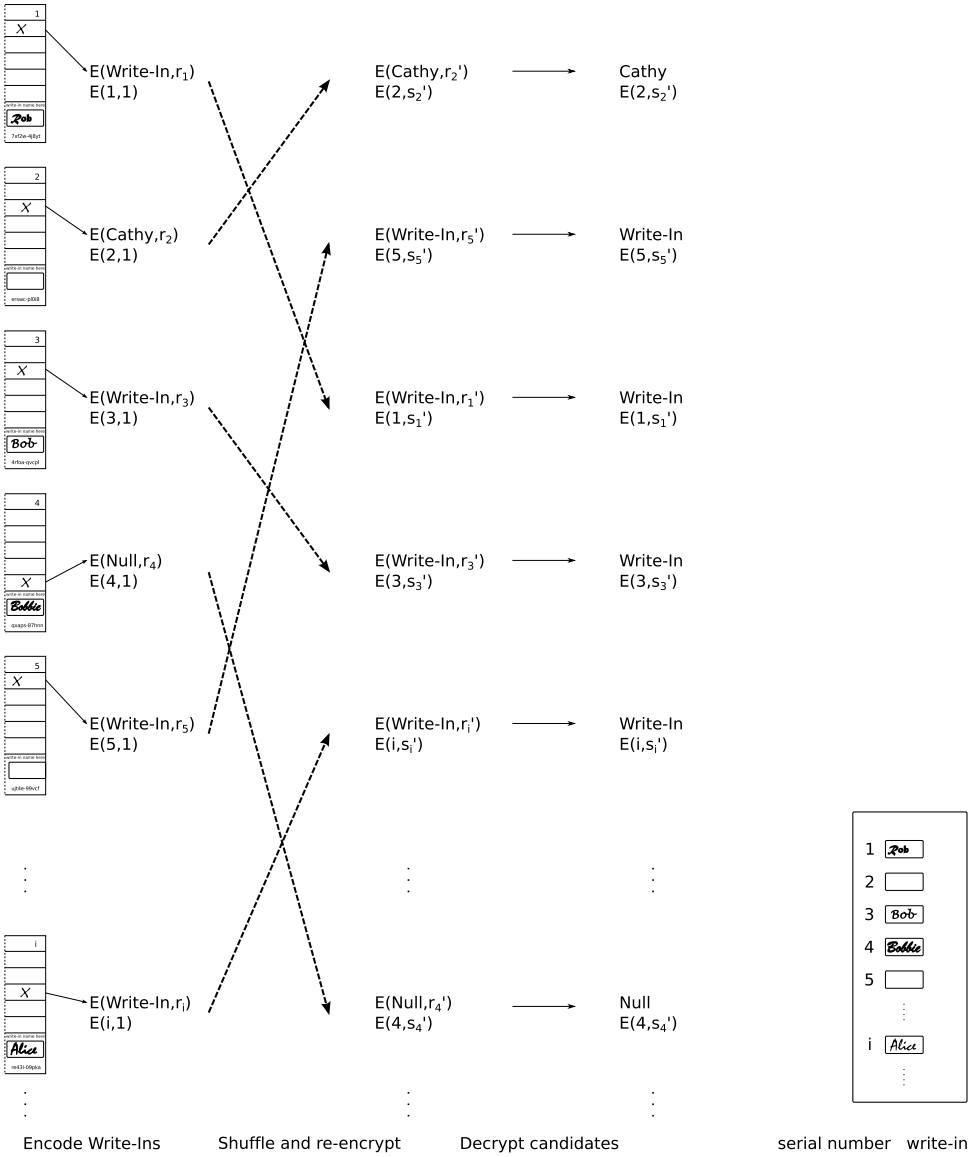

Fig. 5. Mixing the votes

of the listed candidates or a null vote. This process is illustrated in Figure 7. Hence this process does not reveal whether the corresponding vote C_j was counted for the written-in candidate or not (except where this is evident from the result). This prevents the voter from obtaining a receipt for a vote for their write-in candidate.

3.2 Flexibility on reporting write-in results

In practice different electoral regulations specify different rules for allowed write-ins and for reporting on write-ins, giving rise to a whole range of situations that any election system might need to deal with, from reporting tallies for all written-in names, to only reporting on tallies of 'official' write-in candidates (who have registered with the electoral authorities in some way). The output of the first mix (Figure 5) indicates the total number of write-ins; and the scans of the write-ins on the bulletin board, once assigned to candidates, give an upper bound for the number of votes received by each write-in candidate. If it is apparent from these intermediate numbers that the write-in votes cannot make a difference to the result of the election, it may be allowable to report on the result even before the write-ins are tallied.

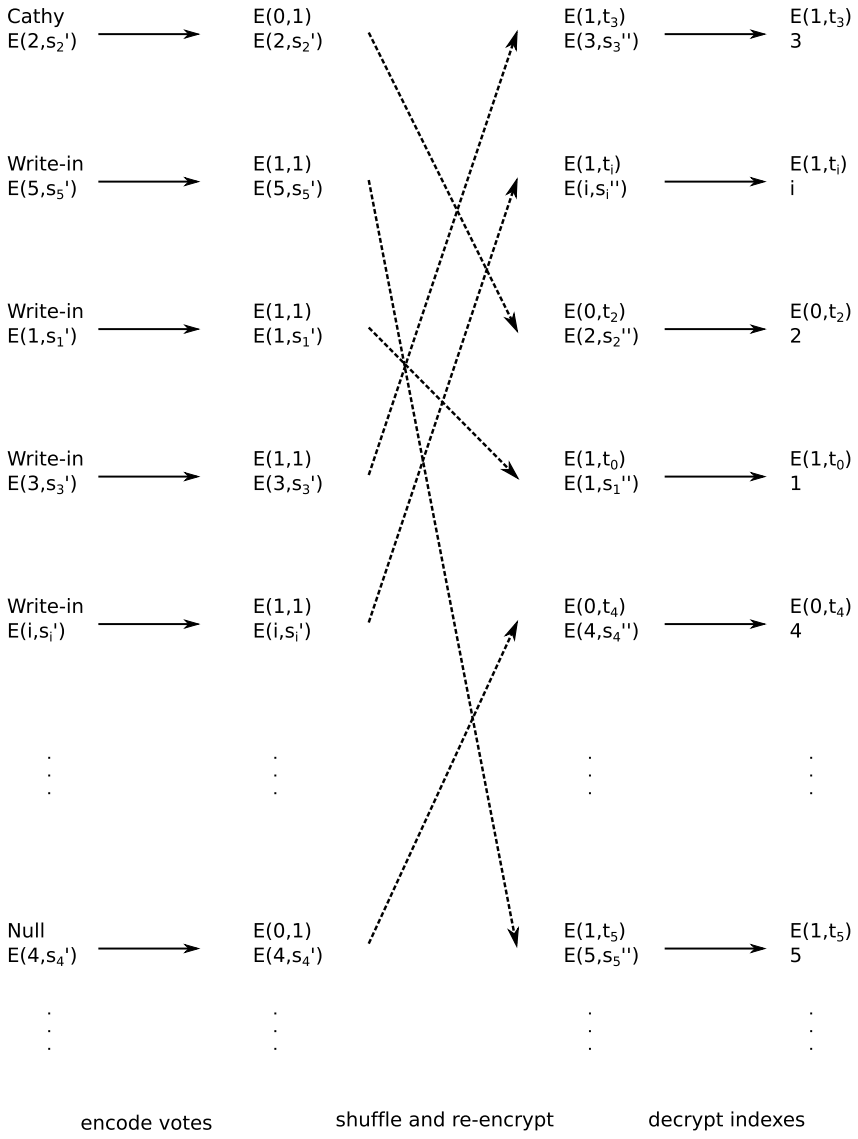

Fig. 6. Extracting the write-ins

The write-ins on the bulletin board can be interpreted and sorted by candidate, and only those to be reported need to be tallied by means of the second mix and homomorphic tallying. Thus the stages of the tallying process can be tailored to the local regulations regarding write-ins.

4 Receipt-freeness

Informally, receipt-freeness is the property that a voter cannot obtain evidence of how they voted. Benaloh and Tuinstra [3] introduced the issue of receipt-freeness, and considered it as the property that does not allows a voter "to prove that a vote was cast in a particular way".

Okamoto [13] gives a formal definition of receipt-freeness, formalising the property as the ability of the voter to cast a vote v_i so that a coercer cannot tell that their preferred vote v_i^* has not been cast. In the case where a receipt is issued, this means that the receipt obtained by the voter following a vote v_i is consistent with a vote for v_i^*. For the purposes of this paper, we will consider what the coercer can learn from the information contained in the receipt retained by the voter.

Even at this level of abstraction, an analysis is useful in the context of write-ins since the asymmetry between listed candidates and write-in candidates already introduces some interesting considerations. A full security analy-

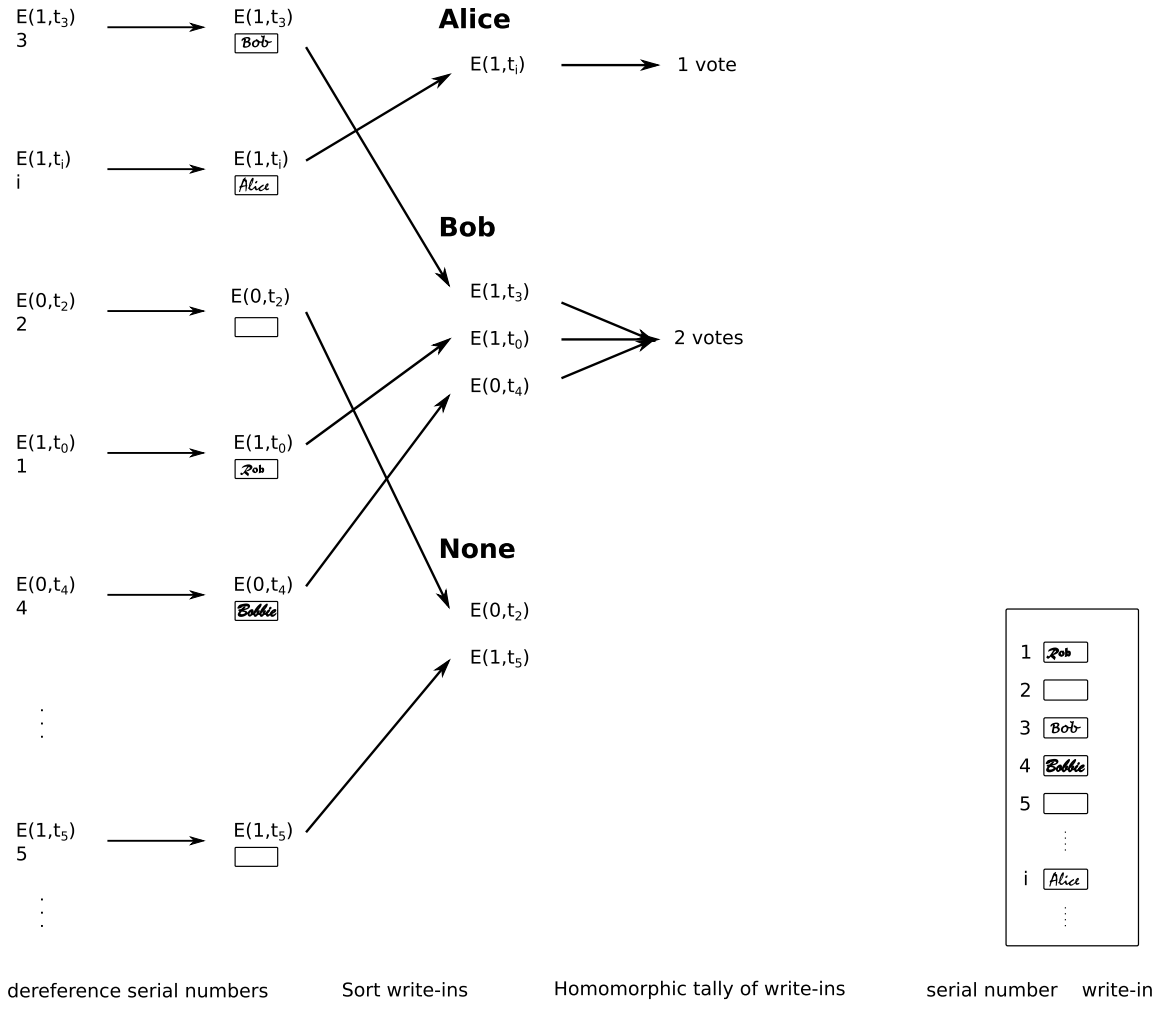

Fig. 7. Tallying the write-ins

sis for receipt-freeness will require more detailed modelling of the system: the Universal Composability approach [**?**] is a good candidate to provide an appropriate framework, as it provides a rigorous model for coercion, incorporates the modelling of the system, and allows for the information revealed by the posting of receipts on the bulletin board and the tallying of the election. This is the subject of ongoing research.

We adapt Okamoto's definition of receipt-freeness to our current setting, referring to receipts explicitly, as follows:

Definition 1 (After Okamoto). *A voting scheme with receipts is* receipt-free *if a coercer who requires the voter to vote for candidate c_i^*, and will accept a receipt r_i^*, cannot tell that the voter's preferred candidate c_i did not receive the vote.*

In other words, any receipt that a coercer will accept is consistent with a vote for any candidate.

Classical Prêt à Voter is receipt-free in this sense: the receipt that the voter obtains could correspond to a vote for any of the candidates, and so it does not provide any evidence of which candidate received the vote, or any useful information about the vote.

In the scheme incorporating write-ins presented in Sections 2 and 3, any receipt retained by a voter is consistent with any valid vote that she may have cast, including a valid vote for any write-in candidate. Since a dummy receipt can be of any form at all, and bears no relationship to the valid vote cast, any receipt a voter might retain is consistent with any valid vote. The scheme thus meets the characterisation of *receipt-freeness* given in Definition 1 above.

4.1 The scheme without dummy ballots

If we restrict the scheme to using only valid ballots, and not dummy ballots, then the receipt-freeness properties are weakened with respect to write-in candidates, and receipts do leak some information:

- A receipt of a valid ballot, with a particular write-in, is not consistent with a vote for a different write-in candidate. It is consistent with a vote for the written-in candidate, and for any of the listed candidates.
- A receipt of a valid ballot with no write-in is not consistent with a vote for any write-in. It is consistent with a vote for any of the candidates, and for a spoiled ballot (i.e. a vote for a write-in but no candidate written-in).

Hence the system without the use of dummy ballots does *not* provide receipt-freeness in the case of write-ins. While any receipt is consistent with any vote for a listed candidate, it is consistent with a vote for a write-in candidate only if that name is written in the write-in box. If receipt r_i^* does not contain a write-in name, or contains a write-in name other than wi_i, then a coercer *can* tell that a voter's preferred (write-in) candidate wi_i did not receive the vote; and thus Definition 1 does not hold.

Observe that the weaker form of Definition 1 below does hold, where the voter's preferred candidate c_i is a listed candidate:

Definition 2 (Receipt-freeness for listed candidates). *A voting scheme with receipts is* receipt-free for listed candidates *if, for any listed candidate c_i, a coercer who requires the voter to vote for candidate c_i^*, and will accept a receipt r_i^*, cannot tell that the voter's preferred listed candidate c_i did not receive the vote.*

The voter may vote for any listed candidate while remaining consistent with any receipt a coercer might require. However, this definition is too weak to give full receipt-freeness: the voter does not have the required guarantees in the case where she wishes to vote for a write-in candidate.

4.2 Optional dummy ballots

Rather than give both a valid and a dummy ballot form to each voter, a suggested approach is to offer the voter the *option* of whether they wish to take a dummy alongside a valid form, rather than making it compulsory. An alternative approach could offer voters the opportunity to take as many dummy ballots as they wished.

By making the choice optional, voters still have the ability to obtain a dummy receipt if required, thus retaining receipt-freeness of the system.

In this case it is essential that a coercer should not know whether or not a voter opted to take the dummy ballot. Otherwise the additional knowledge that a voter did not select a dummy reduces receipt-freeness to the case where dummy ballots are not used. This means that the procedure for offering the voter the choice, and providing the voter with the appropriate ballot form(s), must be carefully designed to prevent information leaking to the coercer.

Dummy votes do not contribute to the final election result, and are provided purely to provide receipt-freeness. Since dummy ballots are processed in exactly the same way as valid ballots, their inclusion introduces a processing overhead, and reducing the number of dummy ballots passed into the system will improve efficiency. Making dummy ballots optional for voters provides a good way of achieving this.

5 Discussion

5.1 Human factors

The inclusion of a write-in option on the ballot form introduces an inherent asymmetry between candidates: write-in candidates are handled differently to listed candidates. While we see that any receipt is compatible with any vote, in practice it may be that voters will use the ballot forms in particular ways. For example it seems plausible that many voters will only complete the write-in box if they are voting for that write-in candidate, and that they will otherwise leave it blank. Such a pattern of behaviour would impact on the information given by a receipt: a receipt including a write-in name might in practice indicate a higher likelihood of a vote for that candidate than a receipt without that name written-in. Empirical investigations would be necessary to gain an understanding of the bias introduced by voter practise.

5.2 Comparison with other schemes

Few proposals for secure election systems explicitly address write-in ballots. For example, the Helios online election system [1] does not address the issue of write-ins explicitly, and the current implementation does not provide for them. The JCJ scheme of [8] excludes write-ins, because of the concern that an attacker could coerce voters to introduce specific strings into the write-in box and check that they have done so. Our scheme minimises the impact of such coercion, since the presence of a write-in on the Prêt à Voter receipt is not evidence of a vote for that write-in candidate.

In this section we will consider two schemes which do allow write-ins: Vector Ballots, and Scantegrity II. We will consider them with respect to the various aspects discussed above.

Vector Ballots

The Vector Ballot approach described [10] is to our knowledge the only cryptographic e-voting protocol designed specifically to support write-in ballots. It is a purely electronic system, intended for internet voting. A vector ballot accepts both write-in candidates and votes for listed candidates. A completed ballot is a vector with three components: an encrypted flag indicating whether the vote is for a write-in or a listed candidate; a ciphertext possibly containing a listed candidate; and a ciphertext possibly containing a write-in. Correctly completed ballots are either

1. an encrypted flag value of 0, an encryption of a selected candidate, and an encryption of 0 for the write-in; or
2. an encrypted flag value of 1, an encryption of 0 for the selected candidate, and an encryption of the write-in candidate name.

The non-write-in ballots can then be tallied using homomorphic encryption. The write-ins cannot be tallied homomorphically since they are not predetermined; they must be extracted and revealed individually in order to enable tallying. This is achieved by a 'shrink and mix' procedure: the sequence of ballots is divided up into batches, and then for each batch, if *any* of the write-in flags are set, then the whole batch is included as input to the mix. However, any batch with no write-ins will not be input to the mix. Hence all write-ins, and a number of non-write-ins, will be input to the mix, but the list of ballots is reduced. The motivation for this is to improve efficiency: mixes are computationally expensive, and since only small proportions of voters typically opt for write-ins, many of the ballots can be excluded from the mix while still masking which of the input ballots are write-ins. The output of the mix can then be decrypted, the null write-ins discarded, and the genuine write-ins tallied.

With regards to receipt-freeness, the only information provided by the system is whether the ballot is input into the mix. If it is input then the ballot could be for a listed candidate or a write-in. However, if it is not included in the input then it is clearly not for a write-in candidate. Hence it is possible (indeed, likely) for voters voting for a listed candidate to obtain evidence that they did not vote for any write-in candidate. However, it is possible for all votes to be put into the mix by considering all the votes as one batch, and not applying the 'shrink' stage of the process. Hence if necessary the scheme can provide receipt-freeness of not having voted for a write-in, though at the cost of efficiency of the tally.

The scheme does not allow decoy write-ins. Although voters' receipts (i.e. what is on the bulletin board as received votes) do not expose write-in candidate entries, they are exposed in the output from the mix, and all such votes will have been cast as write-ins. Since these can in principle contain arbitrary strings, voters can be subjected to the forced abstention attack by being coerced into entering such strings. The appearance of such a string in the mix output provides evidence that the voter did indeed waste their vote.

This last property contrasts with our approach, where the published material does not indicate whether the string was selected as a write-in. If the authorities do not tally the results for such a string, then its appearance does not provide evidence of a wasted vote.

Scantegrity II

The description of Scantegrity II in [6] does not explicitly discuss write-ins, but they were incorporated into the system to meet the election requirements of the Takoma Park municipal election which was run using Scantegrity II. The system takes a different approach to the systems discussed above, in that the voters do not retain any receipt of what they have written in, so they are not able to demonstrate whether they have or have not written-in any particular candidate. In this sense, the Scantegrity II approach provides a stronger form of receipt-freeness than the others with respect to write-ins. In that system, the final tally for 'Write-In' is verifiable, but a level of trust

is required in the election authority, since a malicious authority could change the write-in images[4]. If necessary, additional procedures could be introduced to verify the write-ins. This contrasts with our scheme, which allows voters to confirm that the write-in image they provided has not been changed by virtue of the receipt and the published election data.

6 Summary

We have presented an extension of Prêt à Voter to include a mechanism for allowing write-in candidates. The approach provides the voter with a receipt which can contain an image of the (hand-written) write-in but still provides receipt-freeness, in the sense that the voter's receipt is not evidence of the voter's vote. The scheme provides maximum flexibility when tallying, allowing write-ins to be assessed and assigned against write-in candidate names by election officials before they are tallied, and tallying only when necessary according to local requirements. Homomorphic tallying allows write-ins to be tallied without revealing which particular votes were cast for them, providing secrecy of the ballot. We have discussed various aspects of the property of receipt-freeness with respect to the information contained in the receipt, and seen the need for dummy or null ballots to provide receipt-freeness for write-ins. We believe that the approach taken in this paper and developed for Prêt à Voter will be more generally applicable to other voting schemes in which a voter obtains a receipt of how they marked their ballot form.

Acknowledgements

We are grateful to Ron Rivest, Peter Ryan, and Emily Shen for discussions and comments on these ideas. This work was funded by EPSRC under grant EP/G025797/1, and was conducted while James Heather was a Royal Academy of Engineering/Leverhulme Trust Senior Research Fellow.

References

1. Ben Adida. Helios: Web-based open-audit voting. In Paul C. van Oorschot, editor, *USENIX Security Symposium*, pages 335–348. USENIX Association, 2008.
2. Ben Adida and Ronald L. Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In Ari Juels and Marianne Winslett, editors, *WPES*, pages 29–40. ACM, 2006.
3. Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC*, pages 544–553, 1994.
4. R. Carback, D. Chaum, J. Clark, J. Conway, A. Essex, P.S. Herrnson, T. Mayberry, S. Popoveniuc, R.L. Rivest, E. Shen, A.T. Sherman, and P.L. Vora. Scantegrity II municipal election at Takoma Park: The first E2E binding governmental election with ballot privacy. In *Proceedings of the 19th USENIX Security Symposium*, 2010.
5. David Chaum. Punchscan. `http://www.punchscan.org`. Viewed on 12 November 2010.
6. David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L. Rivest, Peter Y. A. Ryan, Emily Shen, Alan T. Sherman, and Poorvi L. Vora. Scantegrity II: end-to-end verifiability by voters of optical scan elections through confirmation codes. *IEEE Transactions on Information Forensics and Security*, 4(4):611–627, 2009.
7. Markus Jakobsson, Ari Juels, and Ronald L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In Dan Boneh, editor, *USENIX Security Symposium*, pages 339–353. USENIX, 2002.
8. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In Vijay Atluri, Sabrina De Capitani di Vimercati, and Roger Dingledine, editors, *WPES*, pages 61–70. ACM, 2005.
9. Aggelos Kiayias and Moti Yung. The vector-ballot e-voting approach. In Ari Juels, editor, *Financial Cryptography*, volume 3110 of *Lecture Notes in Computer Science*, pages 72–89. Springer, 2004.
10. Aggelos Kiayias and Moti Yung. The vector-ballot approach for online voting procedures. In David Chaum, Markus Jakobsson, Ronald L. Rivest, Peter Y. A. Ryan, Josh Benaloh, Miroslaw Kutylowski, and Ben Adida, editors, *Towards Trustworthy Elections*, volume 6000 of *Lecture Notes in Computer Science*, pages 155–174. Springer, 2010.
11. mfoster.com. 2008 Presidential Election Write-In Rules. `http://mfoster.com/misc/write-in_rules_2008.html`. Viewed on 15 November 2010.
12. C. Andrew Neff. A verifiable secret shuffle and its application to e-voting. In *ACM Conference on Computer and Communications Security*, pages 116–125, 2001.
13. Tatsuaki Okamoto. Receipt-free electronic voting schemes for large scale elections. In Bruce Christianson, Bruno Crispo, T. Mark A. Lomas, and Michael Roe, editors, *Security Protocols Workshop*, volume 1361 of *Lecture Notes in Computer Science*, pages 25–35. Springer, 1997.
14. Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa. Efficient anonymous channel and all/nothing election scheme. In *EUROCRYPT*, volume 765 of *LNCS*, pages 248–259. Springer, 1993.

15. Peter Y. A. Ryan. Prêt à voter with Paillier encryption. Technical Report CS-TR-965, University of Newcastle, 2006.
16. Peter Y. A. Ryan, David Bismark, James Heather, Steve Schneider, and Zhe Xia. Prêt à voter: a voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security*, 4(4):662–673, 2009.
17. Peter Y. A. Ryan and Steve A. Schneider. Prêt à voter with re-encryption mixes. In Dieter Gollmann, Jan Meier, and Andrei Sabelfeld, editors, *ESORICS*, volume 4189 of *Lecture Notes in Computer Science*, pages 313–326. Springer, 2006.
18. Peter Y. A. Ryan and Vanessa Teague. Ballot permutations in Prêt à Voter. In *Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop*, 2009.
19. Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In *EUROCRYPT*, volume 921 of *LNCS*, pages 393–403. Springer, 1995.
20. Zhe Xia, Chris Culnane, James Heather, Hugo Jonker, Peter Ryan, Steve Schneider, and Sriramkrishnan Srinivasan. Versatile Prêt à Voter: Handling multiple election methods with a unified interface. In *Indocrypt: 11th International Conference on Cryptology in India*, 2010.