Mix and Test Counting in Preferential Electoral Systems

Roland Wen Richard Buckland

University of New South Wales, Australia
{rolandw,richardb}@cse.unsw.edu

Technical Report
UNSW-CSE-TR-0809
April 2008

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Although there is a substantial body of work on online voting schemes that prevent bribery and coercion of voters, as yet there are few suitable schemes for counting in the alternative vote and single transferable vote preferential systems. Preferential systems are prone to bribery and coercion via signature attacks. This is an issue for online elections in Australia, where all parliamentary elections use these preferential systems. We present the Mix and Test Counting scheme, a preferential counting protocol that is resistant to signature attacks. For the alternative vote, it reveals no information apart from the identity of the winning candidate. For the single transferable vote, it reveals additional anonymised counting information. However the only candidates identified are the winning candidates.

1 Introduction

Most online voting schemes in the literature are designed for plurality (first past the post) electoral systems, where the winner is the candidate who receives the most votes. But using these schemes in preferential electoral systems exposes voters to potential bribery and coercion through signature attacks. We introduce two related preferential counting protocols that protect voters from such attacks.

Preferential electoral systems are widespread in Australia. Indeed, all Australian parliamentary elections at national and state levels use preferential systems. In most cases, elections for the lower house use the *alternative vote* and elections for the upper house use the *single transferable vote*. These systems are also common in the Republic of Ireland and Malta.

The aim of preferential electoral systems is to give voters greater scope in expressing their choices. The distinguishing feature of these systems is that voters *rank* candidates in order of preference. The alternative vote and single transferable vote are some of the more complex instances of preferential systems because the counting procedure iterates rounds of counting. Each round excludes one or more candidates and transfers the votes for these candidates to the remaining (not yet excluded) candidates according to the preferences listed on their ballots. We elaborate below on the mechanics of the counting procedure in these two systems.

1.1 The Alternative Vote

The alternative vote, also known as preferential voting or instant runoff voting, is a majoritarian system for filling a single vacancy. To be elected, a candidate must receive a *majority* of the votes.

Voters cast ballots by ranking the candidates in order of preference. Starting from 1 for the most preferred candidate, a voter assigns consecutive preferences to every candidate on the ballot. A variant is optional preferences, where voters assign a minimum of one preference but need not assign all preferences.

The counting takes place in rounds. In each round, the election authorities tally the votes for the most preferred remaining candidate in each ballot. If no candidate receives a majority, then the candidate with the lowest round tally is excluded. The authorities transfer each vote for that excluded candidate to the next remaining candidate on the corresponding ballot. This process is then repeated until a candidate is elected.

1.2 The Single Transferable Vote

The single transferable vote, also known as the Hare-Clark system or choice voting, is a proportional system for filling multiple vacancies. To be elected,

a candidate must receive a *quota* of votes. The counting procedure continues until all vacancies are filled.

The single transferable vote is a generalisation of the alternative vote to the case of electing multiple candidates. This leads to further complexity when an elected candidate's votes exceed the quota and there are still remaining vacancies. In this case, the counting process excludes the elected candidate and transfers only the votes surplus to the quota. Single transferable vote systems vary according to how they determine which votes elect the candidate and which votes are surplus.

Many solutions exist for transferring surplus votes, with debatable degrees of fairness. Apart from the random sampling method, all solutions transfer ballots at a fractional value. This paper addresses the weighted inclusive Gregory method, which transfers *all of the ballots* for the elected candidate at a fractional value. Each ballot has an individual weight w, with $w = 1$ initially. The surplus transfer value is $\frac{s}{t}$, where s is the surplus and t is the total value of the votes for the elected candidate. For each ballot, its new weight is $w = \frac{s}{t}w'$, where w' is the old weight.

1.3 The Signature Attack

The information-rich nature of the votes in preferential systems introduces the possibility of the *signature attack*, commonly referred to as the *Italian attack* due to its use in elections in Italy [3]. The signature attack is an effective technique for bribing and coercing voters because it potentially compromises voter anonymity during the counting. It can apply to any election in which the number of improbable voting options is relatively large compared to the number of all possible voting options. Preferential elections are particularly vulnerable because the number of possible preference permutations is factorial in the number of candidates.

To "sign" a preferential ballot, a voter can allocate the first preference to a particular candidate and use the remaining preferences as a covert channel. This channel contains the signature, which is a specified ordering of the remaining candidates. Even for a relatively small number of candidates and a large voting population, this signature is highly likely to be unique. An election with C candidates has $(C-1)!$ possible signatures. The national upper house election in Australia has about 80 candidates, and so there are roughly 79! possible signatures. Even if every atom in the universe voted in this election, there would still be a negligible probability that a randomly chosen signature was not unique.

The covert signature is revealed when the ballots are exposed, and it links the voter with the vote. In traditional paper elections, only election authorities and independent scrutineers appointed by the candidates should learn the contents of the ballots. We can only hope they are trustworthy.

With only partial knowledge of the votes, subtle variations of the sig-

nature attack may still be feasible. An adversary can embed uncommon sequences of preferences in the signatures. Then the adversary can glean any available information about these contrived sequences to narrow down the set of *possible signatures*. Naturally, much depends on the eventual distribution of the votes cast. But the adversary can make some educated guesses, especially when there are few major candidates and many minor candidates. Even when the adversary cannot identify exact signatures in the votes cast, it is still possible to determine that particular signatures are not present.

Ideally, to eliminate the possibility of covert channels and intentional or accidental information leakage, the counting process should be secret and reveal only the final result. If this is not possible, then to minimise the potential for signature attacks the counting should at least avoid linking revealed information to the candidates. The challenge lies in counting the votes in a secret yet publicly verifiable manner.

1.4 Contributions

The first contribution of this paper is to formulate definitions of security for cryptographic preferential counting schemes. We introduce the strong notion of *counting privacy*, meaning that the counting reveals no information apart from the identity of the winning candidates. We also introduce a weaker notion of *signature resistance*, meaning simply that any revealed round tallies are anonymised with respect to the candidates and other rounds.

Then we present the Mix and Test Counting scheme for preferential counting. The alternative vote version achieves counting privacy. The single transferable vote version does not, but it still achieves signature resistance. Both versions also achieve correctness, public verifiability, and robustness against a minority of corrupt authorities.

1.5 Organisation

We start by discussing existing voting schemes and preferential counting schemes (Section 2). We then describe our security model (Section 3) and the necessary cryptographic building blocks (Section 4). Following this, we present the details of the Mix and Test Counting scheme for the alternative vote and single transferable vote, along with modifications for the optional preferences case (Section 5). Finally, we analyse the security and complexity of our scheme (Section 6). We also explain how to construct an online preferential election scheme by combining our counting scheme with existing voting schemes (Appendix A).

2 Related Work

In the general literature on online elections, preventing bribery and coercion centres on the requirements of *receipt-freeness* and *coercion-resistance*. Informally, receipt-freeness means that voters who cast valid votes cannot prove how they voted. Coercion resistance is the stronger requirement that voters cannot even prove that they cast invalid votes or abstained. But in the context of preferential counting, receipt-freeness and coercion resistance are equivalent notions because vote nullifying attacks are issues for the voting process, not the subsequent counting process.

Receipt-free and coercion-resistant voting schemes focus on protecting voters from bribery and coercion during the voting itself. But they rarely consider the details of the counting. During the counting these schemes generally rely on statistical uncertainty in the votes to prevent voters from being identified by their votes. Every possible voting option must be likely to receive some votes from honest voters. For simple plurality elections, this is generally a reasonable assumption. But for preferential elections, it is not. This compromises receipt-freeness and coercion resistance.

Voting schemes have two main approaches to counting votes: public counting and private counting. Both methods reveal all unique votes, and so both are susceptible to signature attacks.

Public counting schemes [9, 10, 13, 14, 20] anonymise the encrypted votes (generally through mix-nets) and then publicly reveal all the votes. Any party can then calculate the result. Conversely, private counting schemes [1, 2, 7, 17] maintain the secrecy of individual votes. The authorities combine all the encrypted votes into encrypted tallies using an additively homomorphic cryptosystem. Then they decrypt only these tallies. But as there is a tally for every possible voting option, this reveals as much as public counting.

To address this, Goh and Golle [4] propose a counting scheme for alternative vote elections. In this scheme, the ballot contains a list of encrypted counters, and each counter represents the preference for a candidate. In each round of counting, the authorities calculate encrypted tallies for the candidates and then decrypt these tallies. To exclude a candidate, the authorities secretly update the counters in every ballot. This scheme severely limits signature attacks because it only discloses the round tallies for each candidate. But one drawback is the high computational and communication cost. In an election with V voters and C candidates, the costs for each authority are $O\left(VC^4\right)$ and the costs for the voter are $O\left(C^4\right)$.

Heather [6] proposes an election scheme for single transferable vote elections using electronic voting devices (Prêt à Voter). The ballot structure is the opposite of that in Goh and Golle's scheme. Here, the ballot is a list of encrypted votes for the candidates in decreasing order of preference. There is also an encrypted null vote to indicate the end of the ballot. In each round

of counting, the authorities decrypt the first vote of each ballot and allocate the ballots to each candidate according to these revealed votes. Then they calculate the tallies simply by counting each candidate's batch of ballots.

To exclude a candidate, the authorities re-allocate that candidate's ballots. In each of these ballots, the authorities move the first encrypted vote (for the excluded candidate) to the end of the ballot. Then they decrypt the next encrypted vote. If the vote is for a remaining candidate, then the authorities allocate the ballot to that candidate. If the vote is null, then the ballot is exhausted and they discard it. Otherwise the vote must be for a previously excluded candidate, and the authorities move that encrypted vote to the end of the ballot and place the ballot aside in an unallocated batch. After transferring all the other ballots, the authorities return to the unallocated batch. They mix this batch using a mix-net (see Section 4.3) and then repeat the transfer process until all ballots are allocated to the remaining candidates. At the end of the round, the authorities mix each candidate's ballots to conceal the correspondence between preferences in a ballot across rounds.

In addition to revealing the candidates' round tallies, the transfer method leaks sequences of preferences for previously excluded candidates. Although the authorities mix the unallocated ballots to conceal the exact sequences, this information is still hazardous especially when transferring a small number of ballots. It facilitates signature attacks by significantly narrowing down the set of possible signatures.

To handle surplus transfers, this scheme uses a novel technique. Consider a surplus transfer value $\frac{s}{t}$. Before transferring any ballots, the authorities make $t-1$ copies of every ballot, giving t copies of each ballot in total. They also multiply the quota by t. Then they transfer s copies of each ballot for the excluded candidate using the above transfer method. This preserves the relative weights of the ballots, albeit at high cost.

This surplus transfer technique avoids tagging each ballot with a public fractional weight because an adversary could use such information to trace the surplus transfer history of these ballots across rounds. Since a surplus transfer value $\frac{s}{t}$ is highly likely to be unique across all surplus transfers, it would serve as a unique tag on each ballot in the surplus transfer (in addition to other unique tags from previous surplus transfers). Then in all subsequent rounds, an adversary could identify the ballots that originated from this surplus transfer. This would further cull the set of possible signatures.

However this technique is not sufficient to prevent an adversary from extracting the information from the round tallies. Suppose in round r a candidate is elected with t_r votes and a surplus s_r. In the next round $(r+1)$, a remaining candidate's round tally will be $t_{r+1} = xt_r + s_r y$, where x is that candidate's round tally before the surplus transfer and y is the number of original surplus ballots (before duplication) received from the elected candidate. Then y can be recovered as $y = (t_{r+1} \bmod t_r) \div s_r$. The

adversary can repeatedly apply this technique to consecutive round tallies for each candidate in order to reconstruct the entire surplus transfer history. Then in any subsequent round, even after additional surplus transfers, the adversary knows the number of original surplus ballots that each candidate currently holds. This provides the same information to the adversary as explicitly identifying the cumulative fractional weight of each ballot.

3 Security Model

We present a security model for preferential counting schemes. The input to a counting scheme is a list of valid ballots and a list of valid candidates. We assume that the number of ballots is large compared to the number of candidates. The output is the identities of the winning candidates.

3.1 Participants and Adversary Model

The only participants are the authorities who perform the counting. All communication is public and via an authenticated bulletin board. We consider a static, active adversary who can corrupt up to a threshold of the authorities.

3.2 Security Requirements

We introduce two alternative requirements for preferential counting schemes.

Definition 3.1 (Counting Privacy)**.** Apart from the identities of the winning candidates, the counting scheme reveals no information.

Definition 3.2 (Signature Resistance)**.** Apart from the round tallies and the identities of the winning candidates, the counting scheme reveals no information. The round tallies are anonymised with respect to the candidates and other rounds.

Counting privacy applies to counting in any electoral system. It provides the strongest possible protection against all potential counting attacks.

Signature resistance applies specifically to preferential systems that iterate rounds of counting. It provides strong protection against signature attacks. For example, it prevents the standard signature attack. The information revealed by a signature-resistant counting scheme is a strict subset of that revealed by traditional paper elections and the existing schemes described in Section 2, all of which publicly release identifiable round tallies.

The remaining security requirements are familiar for online elections.

Definition 3.3 (Correctness)**.** All valid votes are correctly counted and no invalid votes are counted.

Definition 3.4 (Public Verifiability)**.** Any observer can check that the result is correct.

Definition 3.5 (Robustness)**.** The counting cannot be disrupted or compromised by a minority of corrupt authorities.

Counting schemes do not consider requirements that only relate to voters during the preceding voting phase, for example voter eligibility, individual verifiability and robustness with respect to corrupt voters.

3.3 A Caveat for Real World Elections

The Australian Electoral Commission and its state counterparts publish detailed counting statistics in the official results for every election. Consequently, counting privacy and signature resistance ensure a higher level of counting secrecy than Australian elections. But even though public verifiability provides strong confidence in the result, an important question is whether the voting population would be willing to accept complete secrecy. At the very least, it could prove unpopular among political commentators, and politicians planning future election campaigns! Moreover, there are unavoidable circumstances that require the disclosure of certain counting data. For example, the Australian Electoral Commission provides public funding to candidates depending on the number of first preference votes a candidate receives in an election.

Cryptographic counting schemes that initially satisfy counting privacy or signature resistance could be deliberately weakened to reveal additional information if necessary. But of course this could compromise security if the public results themselves provide sufficient information for mounting large scale signature attacks. Precisely how much information is too much is an open problem.

4 Cryptographic Primitives

The Mix and Test Counting scheme relies on three cryptographic primitives: an additive and scalar multiplicative homomorphic threshold cryptosystem, plaintext equality and inequality tests, and mix-nets. Rather than depending on specific instances of these primitives we model them as ideal primitives.

4.1 Additive and Scalar Multiplicative Homomorphic Threshold Cryptosystem

An additive and scalar multiplicative homomorphic cryptosystem [15] is a probabilistic public-key cryptosystem with the following properties:

1. Given two encrypted messages $E(m_1), E(m_2)$, anyone can compute $E(m_1) \times E(m_2) = E(m_1 + m_2)$.

2. Given an encrypted message $E(m)$ and a scalar s, anyone can compute $E(m)^s = E(sm)$.

In the threshold version of such a cryptosystem, each authority has a secret share of the private key such that a quorum of authorities must collaborate to decrypt ciphertexts. The decryption process is publicly verifiable and reveals no information to any coalition of authorities smaller than the quorum.

4.2 Plaintext Equality and Inequality Tests

Plaintext equality and inequality tests compare the plaintexts of given ciphertexts in a publicly verifiable manner without revealing the plaintexts. Given a pair of encrypted messages $E(m_1)$ and $E(m_2)$, a plaintext equality test [8] determines whether $m_1 = m_2$, and a plaintext inequality test [16, 18] determines whether $m_1 \geq m_2$. The tests reveal only the boolean result. Neither test reveals any information to any coalition of authorities smaller than the quorum.

4.3 Mix-nets

A mix-net [5, 12] is a series of servers that each randomly mix (by permuting and re-encrypting) a batch of messages. As long as at least one mix server is honest, the process conceals the correspondence between input and output messages. The mixing is publicly verifiable.

In the case that a message is a vector of ciphertexts, such as with preferential ballots, the mix-net re-encrypts every ciphertext individually and preserves the structure of each vector.

5 The Mix and Test Counting Scheme

The Mix and Test Counting scheme implements preferential counting for the alternative vote and the single transferable vote. Both versions protect voters from the signature attack, with the alternative vote version achieving counting privacy and the single transferable vote version achieving signature resistance.

Mix and Test Counting commences after the voting has finished and the authorities have performed all necessary ballot processing (including removing invalid ballots). The input is a list of valid ballots and a list of valid candidates. The output is the identities of the winning candidates. The authorities post the result of every operation on an authenticated bulletin board with full revision tracking. Whenever the authorities generate encrypted messages, they do so in a verifiable manner.

Each ballot is a vector of encrypted votes in decreasing order of preference. Each vote is probabilistically encrypted with a homomorphic threshold cryptosystem using the public key of the authorities. Initially, every ballot contains a vote for each candidate. In an election for C candidates, the input ballots are of the form $\langle E(v_1), \ldots, E(v_C)\rangle$.

Mix and Test Counting iterates rounds of counting until the election is over. In each round, the authorities consider the first encrypted vote $E(v_1)$ of each ballot. This corresponds to the most preferred remaining candidate in the ballot. The authorities use the *private tallying protocol* in Section 5.3 to secretly tally these encrypted votes. When the authorities exclude a candidate c_i, they use the *private deletion protocol* in Section 5.4 to secretly remove the encrypted vote for c_i from every ballot. Both the private tallying and private deletion protocols maintain complete privacy of the ballots, tallies, and identities of the candidates.

We now present the details of the counting scheme for the alternative vote and the single transferable vote. We then describe the private tallying and private deletion protocols. We also provide modifications to the counting scheme for the optional preferences variant.

5.1 Counting the Alternative Vote

For the alternative vote version of Mix and Test Counting, the authorities perform a complete transfer of the votes by excluding a candidate in every round until there are only two remaining candidates. Performing only exclusions avoids revealing the round in which the winning candidate is elected. Deferring the announcement of the winner does not affect the result. At the end, the authorities reveal the identity of the winning candidate.

At the start of the counting, the authorities initialise the list $\mathbb{R}$ of anonymous encrypted remaining candidates. Initially, $E(c_i) \in \mathbb{R}$ for every candidate c_i. The authorities then mix $\mathbb{R}$ to anonymise the candidates. As candidates are excluded, the authorities update $\mathbb{R}$ so that it only contains the remaining candidates.

The authorities perform each round of counting as follows:

1. Execute the private tallying protocol with $\mathbb{R}$ and the list of ballots. The output is the list

$$\mathbb{C} = \{\langle E(c_i), E(t_i)\rangle\}$$

   of encrypted anonymous counters, where c_i is a remaining candidate and t_i is the round tally for c_i.

2. If there are only two remaining candidates, then immediately stop the counting rounds. Otherwise, determine the minimum counter

$\langle E(c_{min}), E(t_{min})\rangle \in \mathbb{C}$ by executing plaintext inequality tests to compare the pair of tallies $E(t_i)$ and $E(t_j)$ for all the counters

$$\langle E(c_i), E(t_i)\rangle, \langle E(c_j), E(t_j)\rangle \in \mathbb{C}.$$

By tracking the current minimum encrypted tally, this requires $|\mathbb{C}|-1$ comparisons for $|\mathbb{C}|$ counters. Then c_{min} is the excluded candidate.

3. Execute the private deletion protocol with the encrypted excluded candidate $E(c_{min})$ and the list of ballots to remove the encrypted vote for c_{min} from every ballot.

4. Update $\mathbb{R}$ by setting $\mathbb{R} = \{E(c_i)\}$ for all $\langle E(c_i), E(t_i)\rangle \in \mathbb{C}$ such that $E(c_i) \neq E(c_{min})$.

5. Mix $\mathbb{R}$ to conceal the partially revealed ordering of the encrypted candidates (according to the tally comparisons in Step 2).

The authorities repeat these steps until there are only two remaining candidates c_1 and c_2 with counters $\langle E(c_1), E(t_1)\rangle$ and $\langle E(c_2), E(t_2)\rangle$. To determine which candidate c_i is elected, the authorities execute plaintext inequality tests to compare $E(t_1)$ and $E(t_2)$ with an encrypted majority quota $E(q)$. Then they decrypt $E(c_i)$ and reveal the winner. The counting reveals no other information.

5.2 Counting the Single Transferable Vote

For the single transferable vote version of Mix and Test Counting, we make several minor modifications to the alternative vote version in order to account for fractional ballot weights in surplus transfers. We use the same idea as Heather [6] to maintain relative *integer* ballot weights, but we avoid altering the number of ballots. Instead, each ballot has an encrypted integer weight $E(w)$ and the authorities use the scalar multiplicative homomorphic property of the cryptosystem to privately update the ballot weights. If an elected candidate receives exactly the quota of votes, then there is no transfer. In this case, the authorities set $w = 0$ in each ballot for this candidate.

The counting iterates rounds until all vacancies are filled. At the end, the authorities reveal the identities of the winning candidates.

At the start of the counting, the authorities initialise the list $\mathbb{R}$ of anonymous encrypted remaining candidates as before. They also initialise a list $\mathbb{E}$ of encrypted elected candidates to be empty, and a quota q. Then the authorities append an encrypted integer weight $E(w)$ to each ballot. Initially, $w = 1$.

The authorities perform each round of counting as follows:

1. Execute the private tallying protocol with $\mathbb{R}$ and the list of ballots. The output is the list

$$\mathbb{C} = \{\langle E(c_i), E(t_i)\rangle\}$$

of encrypted anonymous counters, where c_i is a remaining candidate and t_i is the round tally for c_i.

2. Initialise a list $\mathbb{X}$ of excluded candidate counters to be empty. Then determine the counters for any elected candidates by executing plaintext inequality tests to compare the encrypted quota $E(q)$ with the encrypted tally $E(t_i)$ for each $\langle E(c_i), E(t_i)\rangle \in \mathbb{C}$. Add each elected candidate counter $\langle E(c_i), E(t_i)\rangle$ to $\mathbb{X}$ and add each corresponding $E(c_i)$ to $\mathbb{E}$. If there are no remaining vacancies, then immediately stop the counting rounds. If no candidate was elected, then find the minimum counter $\langle E(c_{min}), E(t_{min})\rangle$ as in Section 5.1 Step 2 and add it to $\mathbb{X}$.

3. If candidates are elected in the previous step, then use each

$$\langle E(c_i), E(t_i)\rangle \in \mathbb{X}$$

to update the ballot weights as follows:

   (a) Decrypt and reveal t_i, and calculate the surplus $s = t_i - q$. Update the quota as $q = t_i q'$, where q' is the old quota.

   (b) For each ballot, execute plaintext equality tests to compare $E(c_i)$ with $E(v)$, where $E(v)$ is the first encrypted vote in the ballot. If $v = c_i$, then update the ballot weight by computing $E(w)^s = E(sw)$. Otherwise, update the ballot weight by computing $E(w)^{t_i} = E(t_i w)$.

   (c) Mix all the ballots to conceal the surplus ballots.

4. For each encrypted excluded candidate $E(c_i)$, execute the private deletion protocol with $E(c_i)$ and the list of ballots to remove the encrypted vote for c_i from every ballot.

5. Update $\mathbb{R}$ by setting $\mathbb{R} = \{E(c_i)\}$ for all $\langle E(c_i), E(t_i)\rangle \in \mathbb{C}$ such that $\langle E(c_i), E(t_i)\rangle \notin \mathbb{X}$.

6. Mix $\mathbb{R}$ to conceal the partially revealed ordering of the encrypted candidates (according to the tally comparisons in Step 2).

After all vacancies are filled, the authorities mix $\mathbb{E}$ to conceal the round in which each candidate was elected. Then they decrypt each $E(c_i) \in \mathbb{E}$ and reveal the winners.

5.3 Private Tallying Using Anonymous Counters

The private tallying protocol calculates tallies for each candidate without revealing the tallies, the candidates, or the contents of the ballots. The authorities use the additive homomorphic property of the cryptosystem to privately add encrypted votes to encrypted tallies. The inputs are a list of encrypted candidates and a list of ballots. The output is a list of encrypted candidate-tally pairs.

At the start of the private tallying, the authorities create the list $\mathbb{C}$ of anonymous candidate counters. Each counter in $\mathbb{C}$ is an encrypted pair $\langle E(c_i), E(t_i)\rangle$, where $E(c_i)$ is an encrypted candidate from the input candidate list and t_i represents the tally of votes for the candidate c_i. Initially $t_i = 0$ and at the end t_i is the final tally. The authorities mix $\mathbb{C}$ to anonymise the counters.

For the first encrypted vote $E(v)$ of each ballot in the input list, the authorities update the anonymous counter $\langle E(c_i), E(t_i)\rangle$ such that $c_i = v$.

1. Locate $\langle E(c_i), E(t_i)\rangle$ by executing plaintext equality tests to compare $E(v)$ with $E(c_j)$ for each $\langle E(c_j), E(t_j)\rangle \in \mathbb{C}$.

2. For the alternative vote, increment t_i by computing

$$E(t_i) \times E(1) = E(t_i + 1).$$

   For the single transferable vote, update t_i with the ballot's encrypted weight $E(w)$ by computing

$$E(t_i) \times E(w) = E(t_i + w).$$

3. Mix $\mathbb{C}$ to conceal the updated counter.

The authorities repeat these steps for all ballots. The output is the list $\mathbb{C}$.

5.4 Private Deletion Using Cyclic Shifts

The private deletion protocol removes a given encrypted candidate from every ballot without revealing the candidate or the contents of the ballots. The inputs are the encrypted candidate $E(x)$ to delete, and a list of ballots. Every input ballot contains an encrypted vote for x. The output is the list of ballots with those encrypted votes removed.

The aim is to remove $E(x)$ without revealing its position in any ballot. To conceal the position of $E(x)$, the authorities perform cyclic shifts of each ballot and then mix all the ballots. Then the authorities use plaintext equality tests to locate $E(x)$ in each shifted ballot. Although the position of $E(x)$ is known at the instant of removal, it is impossible to correlate this with the position of $E(x)$ in the original ballot ordering. In the case

of weighted ballots, the cyclic shifts disregard the encrypted weight $E(w)$, which always remains at the end of the ballot.

Before performing the shifts, the authorities insert an encrypted marker vote at the start of each ballot. This marker enables the authorities to later return the ballot to its original order. To ensure that the cyclic shifts are uniformly distributed, the authorities add padding ballots so that every possible cyclic shift occurs with exactly the same frequency.

The authorities remove $E(x)$ from the ballots as follows:

1. Create an encrypted marker vote $E(m)$, where m is an invalid vote. For each ballot in the input list, insert $E(m)$ at the front of the ballot.

2. Generate $p = (n - (b \bmod n)) \bmod n$ padding ballots, where b is the number of ballots and n is the number of encrypted votes (including the encrypted marker) in each of the ballots. Hence n divides evenly into the total number of ballots $(b + p)$. Create an encrypted dummy vote $E(d)$, where d is an invalid vote with $d \neq m$. This allows the authorities to later remove the padding ballots. Construct each padding ballot as follows:

   (a) Insert $E(m)$ at the front of the ballot. Then append $E(x)$.

   (b) Append a list of $(n - 2)$ copies of $E(d)$.

3. Concatenate the list of genuine ballots and the list of padding ballots. Then mix all the ballots.

4. For each ballot, perform a cyclic shift by $s = i \bmod n$ on the encrypted votes, where i is the ballot's position in the list of ballots. Then mix all the shifted ballots to conceal the shift used for each ballot.

5. For each ballot $\langle E(v_1), \ldots, E(v_n) \rangle$, locate the encrypted vote $E(v_i)$ with $v_i = x$ by executing plaintext equality tests to compare $E(x)$ with $E(v_j)$ for $1 \leq j \leq n$. Remove $E(v_i)$ from the ballot.

6. Generate another $p' = (n' - (b' \bmod n')) \bmod n'$ padding ballots, where $b' = b + p$ is the current number of ballots and $n' = n - 1$ is the current number of encrypted votes in each of the ballots. Hence n' divides evenly into the new total number of ballots $(b' + p')$. Construct each padding ballot as follows:

   (a) Insert $E(m)$ at the front of the ballot.

   (b) Append a list of $(n' - 1)$ copies of the same $E(d)$ as in Step 2.

7. Concatenate the list of ballots and the list of extra padding ballots. Then mix all the ballots.

8. For each ballot, perform a cyclic shift by $s = i \bmod n'$ on the encrypted votes, where i is the ballot's position in the list of ballots. Then mix all the shifted ballots to conceal the shift used for each ballot.

9. For each ballot $\langle E(v_1), \ldots, E(v_{n-1})\rangle$, locate the encrypted marker vote $E(v_i)$ by executing plaintext equality tests to compare $E(m)$ with $E(v_j)$ for $1 \leq j \leq n-1$. Remove $E(v_i)$ from the ballot and shift the ballot back to its original ordering.

10. Mix all the ballots. Then for each ballot, execute a plaintext equality test to compare the first encrypted vote $E(v_1)$ with $E(d)$. If $v_1 = d$, then this is a padding ballot so remove it from the ballot list.

The output is this final list of ballots.

5.5 Optional Preferences

A common variation in preferential systems is that voters are only required to assign one preference, and the remaining preferences are optional. Mix and Test Counting can also accommodate this situation. In this case, every ballot simply contains an encrypted null vote $E(0)$ as a terminator after the last desired preference. We still require that ballots contain an encrypted vote for each candidate. The voter, or possibly the voting application, enters the remaining preferences in arbitrary order after $E(0)$.

The only change to the counting scheme is in the private tallying protocol. To conceal exhausted ballots, the authorities must tally them as normal ballots. So before the tallying starts, the authorities add a null counter $\langle E(0), E(t_0)\rangle$ to the list $\mathbb{C}$ of anonymous candidate counters. But the counting scheme must disregard the null vote tally in order to avoid excluding the null candidate. So at the end of the tallying, the authorities remove $\langle E(0), E(t_0)\rangle$ from $\mathbb{C}$ by executing plaintext equality tests.

6 Analysis

We now discuss the security and complexity of the Mix and Test Counting scheme.

6.1 Security Requirements

For the most part the security of the Mix and Test Counting scheme follows directly from the properties of the underlying cryptographic primitives. The exception is the private deletion protocol (Section 5.4), which relies on cyclic shifts in Steps 4 and 8 to conceal the positions of the removed votes in each ballot.

To ensure that every possible cyclic shift occurs with exactly the same frequency, the authorities add padding ballots. Using plaintext equality tests to later remove the padding ballots reveals no information about the genuine ballots. Mixing the ballots before and after the shifts ensures that ballots cannot be identified.

After mixing the shifted ballots, using plaintext equality tests to locate the position of a specific vote to remove from a ballot reveals no information about the other votes in the ballot. It only reveals the position of the vote at the point of deletion. The following lemma shows that there is no correlation between this position and the original position of the vote before the shift.

Lemma 6.1 (Statistical Independence)**.** *For any encrypted vote in any ballot its position before a shift and its position after a shift are statistically independent.*

Proof. Let v be a vote in a ballot of n votes. Initially, v is in some position $0 \leq i \leq n-1$. Then v is shifted by $[0, n-1]$ places to some position $0 \leq j \leq n-1$. Let

$P(b_i)$ be the probability v was in position i before the shift,

$P(a_j)$ be the probability v is in position j after the shift,

$P(b_i|a_j)$ be the probability v was in position i before the shift, given v is in position j after the shift,

$P(a_j|b_i)$ be the probability v is in position j after the shift, given v was in position i before the shift.

By construction, every shift offset is exactly equally likely. So

$$P(a_j|b_i) = \frac{1}{n}$$

Then

$$\begin{aligned} P(a_j) &= \sum_{k=0}^{n-1} P(a_j|b_k)\, P(b_k) \\ &= \sum_{k=0}^{n-1} \frac{1}{n} P(b_k) \\ &= \frac{1}{n} \sum_{k=0}^{n-1} P(b_k) \\ &= \frac{1}{n} \end{aligned}$$

Hence

$$P(a_j|b_i) = P(a_j) \tag{6.1}$$

By Bayes' Theorem

$$\begin{aligned} P(b_i|a_j) &= \frac{P(a_j|b_i)\,P(b_i)}{P(a_j)} \\ &= \frac{\frac{1}{n}P(b_i)}{\frac{1}{n}} \\ &= P(b_i) \end{aligned} \tag{6.2}$$

Therefore by (6.1) and (6.2), $P(b_i)$ and $P(a_j)$ are statistically independent. □

Corollary 6.2. *The private deletion protocol reveals no information about the locations of individual votes within a ballot.*

Remark 6.3. Lemma 6.1 provides a perfectly secure reduction to the underlying cryptographic primitives.

We state the security theorems here and provide the straightforward proofs in the full version of the paper.

Theorem 6.4 (Secure Alternative Vote Counting)**.** *Mix and Test Counting achieves counting privacy, correctness, public verifiability and robustness for the alternative vote.*

Theorem 6.5 (Secure Single Transferable Vote Counting)**.** *Mix and Test Counting achieves signature resistance, correctness, public verifiability and robustness for the single transferable vote.*

In addition to satisfying signature resistance, the single transferable vote version of Mix and Test Counting has promising potential to resist undiscovered counting attacks. Compared to the ideal alternative vote version, it reveals only slightly more information. The single transferable vote version reveals the anonymised round tallies only when candidates are elected, except for in the final round. This is done to perform surplus transfers in a practical, publicly verifiable and robust manner. An additive and multiplicative homomorphic cryptosystem would make surplus transfers possible without revealing *any* round tallies. But no currently known cryptosystem with these properties is secure (see for example Wagner [19]).

Using the anonymised round tallies, an adversary can reconstruct a partial transfer history for some anonymous elected candidate using the method we described on Heather's surplus transfer technique in Section 2. However in this case the attack is quite limited because the round tallies are anonymised and only known for elected candidates. So for any revealed round tally, the adversary cannot distinguish between whether the candidate received votes directly from the previous surplus transfer or indirectly via intermediate normal transfers. The adversary can only determine the

total number of votes transferred (directly or indirectly) from each previously elected (anonymous) candidate, and the number of full value ballots for this candidate.

6.2 Complexity

We provide estimates of the computational and communication complexity of Mix and Test Counting using typical costs of the underlying cryptographic primitives. As the number of modular operations performed has the same asymptotic complexity as the number of bits transferred, the computational complexity discussion below also refers to the communication complexity.

We consider an election with V voters and C candidates. In each round of counting, the dominating operations in the complexity are the mixing and plaintext equality tests, which each costs $O(VC)$. The number of rounds is $O(C)$, and so the total cost for each authority is $O(VC^2)$. The corresponding complexity for Goh and Golle's alternative vote scheme [4] is $O\left(VC^4\right)$.

It is important that the cost for the voter is not too onerous when combining Mix and Test Counting with a voting scheme (see Appendix A). We assume that the voting scheme requires the voter to provide proofs of correctness at a cost of $O(C)$ for each sub-ballot. As the voter casts C sub-ballots, the total cost is $O(C^2)$. The cost in Goh and Golle's scheme is $O\left(C^4\right)$.

In practice there is an upper limit to C. Otherwise voting can become an onerous task. In traditional elections, the size of the ballot paper is also a physical limitation. Elections in the Australian state of New South Wales are infamous for their cumbersome ballots, with over 300 candidates in upper house elections. This is probably a reasonable approximate upper limit for C.

7 Conclusion

We introduce the Mix and Test Counting scheme for secure counting in preferential elections. It achieves counting privacy for the alternative vote and signature resistance for the single transferable vote. Thus it thwarts known signature attacks for bribery and coercion. The security of this scheme exceeds that of contemporary counting schemes and counting in traditional paper elections. Mix and Test Counting can function as a standalone counting scheme or can be combined with an online voting scheme to form a complete online preferential election scheme.

The private tallying technique has the potential to be applied to existing non-preferential voting schemes as well. Voting schemes that perform public counting can use private tallying and plaintext inequality tests to

privately count the votes. Such elections would reveal only the identities of the winning candidates and thus satisfy counting privacy. Voting schemes that perform private counting and produce a separate encrypted tally for each candidate can achieve the same result using only plaintext inequality tests on the encrypted tallies.

7.1 Future Directions

Australia is beginning to adopt electronic processes for elections. In the short term, Australia is conducting online election trials for military personnel deployed overseas. But many elections in Australia and overseas already use electronic counting, where election authorities enter votes on physical ballots into an electronic database, and a computer calculates the result without cryptographic safeguards. This shift towards naive electronic counting is an alarming trend. It is the worst of both worlds, exposing the election to new risks, but without any protection.

One serious issue with naive electronic counting is the lack of any verifiability due to the difficulty in detecting flaws in the software implementation. Another serious concern is the violation of the secret ballot. Signature attacks are already possible with physical ballots, but they are a greater risk with electronic data. Compromising an electronic database of plaintext votes opens the door to large scale bribery and coercion of voters through signature attacks. Therefore cryptographic approaches to counting have the potential to play an important role as Australia updates its election process.

There is great diversity in electoral systems for democratic elections, but some still have no corresponding cryptographic counting solution. As new democracies develop and existing democracies embrace electoral reform, the variations in electoral systems will no doubt continue to increase. Consequently, there is great scope for future work on counting schemes for complex electoral systems.

Bibliography

[1] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *PODC*, pages 274–283, 2001.

[2] Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC*, pages 544–553, 1994.

[3] Roberto Di Cosmo. On privacy and anonymity in electronic and non electronic voting: the ballot-as-signature attack. `http://www.pps.jussieu.fr/~dicosmo/E-Vote/`, 2007.

[4] Eu-Jin Goh and Philippe Golle. Event driven private counters. In Andrew S. Patrick and Moti Yung, editors, *Financial Cryptography*, volume 3570 of *Lecture Notes in Computer Science*, pages 313–327. Springer, 2005.

[5] Jens Groth. A verifiable secret shuffle of homomorphic encryptions. In Yvo Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 145–160. Springer, 2003.

[6] James Heather. Implementing stv securely in pret a voter. In *CSF*, pages 157–169. IEEE Computer Society, 2007.

[7] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 539–556. Springer, 2000.

[8] Markus Jakobsson and Ari Juels. Mix and match: Secure function evaluation via ciphertexts. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 162–177. Springer, 2000.

[9] 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.

[10] Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang, and Seungjae Yoo. Providing receipt-freeness in mixnet-based voting protocols. In Jong In Lim and Dong Hoon Lee, editors, *ICISC*, volume 2971 of *Lecture Notes in Computer Science*, pages 245–258. Springer, 2003.

[11] Toru Nakanishi and Yuji Sugiyama. Anonymous statistical survey of attributes using distributed plaintext membership test. *Transactions of Information Processing Society of Japan*, 43(8):2414–2424, 2002.

[12] Lan Nguyen, Reihaneh Safavi-Naini, and Kaoru Kurosawa. Verifiable shuffles: a formal model and a paillier-based three-round construction with provable security. *Int. J. Inf. Sec.*, 5(4):241–255, 2006.

[13] Valtteri Niemi and Ari Renvall. How to prevent buying of votes in computer elections. In Josef Pieprzyk and Reihaneh Safavi-Naini, editors, *ASIACRYPT*, volume 917 of *Lecture Notes in Computer Science*, pages 164–170. Springer, 1994.

[14] 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.

[15] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.

[16] David C. Parkes, Michael O. Rabin, Stuart M. Shieber, and C. A. Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. In Mark S. Fox and Bruce Spencer, editors, *ICEC*, volume 156 of *ACM International Conference Proceeding Series*, pages 70–81. ACM, 2006.

[17] Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In *EUROCRYPT*, pages 393–403, 1995.

[18] Berry Schoenmakers and Pim Tuyls. Efficient binary conversion for paillier encrypted values. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 522–537. Springer, 2006.

[19] David Wagner. Cryptanalysis of an algebraic privacy homomorphism. In Colin Boyd and Wenbo Mao, editors, *ISC*, volume 2851 of *Lecture Notes in Computer Science*, pages 234–239. Springer, 2003.

[20] Stefan G. Weber, Roberto Araujo, and Johannes Buchmann. On coercion-resistant electronic elections with linear work. In *ARES*, pages 908–916. IEEE Computer Society, 2007.

A Integration with Voting Schemes

We can construct a solution for online preferential elections by combining the Mix and Test Counting scheme with an existing receipt-free or coercion-resistant voting scheme. A voting scheme is suitable if it satisfies the following properties:

1. Each voter submits a ballot that corresponds to a vote for a single candidate.

2. The scheme produces a public list of encrypted votes.

3. The encrypted votes are compatible with the cryptographic primitives in Section 4.

Although some voting schemes process the ballots to extract the encrypted votes, in most schemes the ballot is already an encrypted vote as we require.

To simplify the discussion, we assume that voters submit encrypted votes, which we call "sub-ballots". A ballot is a vector of sub-ballots.

A preferential election starts with a voting stage using the voting scheme. Then the authorities verify the ballots to remove all invalid ballots. Finally, Mix and Test Counting calculates the election result.

A.1 Voting

For an election with C candidates, the voting scheme conducts C sub-elections in parallel. Each voter casts a ballot consisting of C sub-ballots in decreasing order of preference. Each sub-ballot is encrypted using the public key of the authorities.

When processing the ballots, the authorities preserve the ordering of the sub-ballots in each ballot. After the voting has finished, there is a list of ballots in the form required by Mix and Test Counting.

A.2 Ballot Verification

Before the counting starts, the authorities ensure that every ballot contains a vote for each candidate. To do this they use plaintext set membership tests [11]. Given an encrypted message $E(m)$ and a set of plaintexts $S = \{m_1, m_2, \ldots, m_n\}$, a plaintext set membership test determines whether $m \in S$. The test is publicly verifiable and reveals no information apart from the boolean result. We will make use of the additional property that simultaneously testing multiple encrypted messages for set membership also reveals which encrypted messages correspond to the same plaintext.

For each ballot, the authorities execute a plaintext set membership test to compare all its encrypted votes with the set of valid candidates. The ballot will be valid if all the votes are distinct and for valid candidates. In the case of optional preferences, the set of valid candidates also includes the null vote. It is not strictly necessary to check that the first encrypted vote is non-null because Mix and Test Counting as it stands disregards such ballots.

Note that because verifying the ballots is public, there is (limited) scope for an adversary to construct a signature for an invalid ballot, for example a ballot in which all votes are for the same candidate. To ensure coerced voters can cast additional invalid ballots with fake credentials in coercion-resistant voting schemes [9, 20], the authorities must verify all the ballots before removing any fake ballots.