Partial Disclosure of Votes in STV Elections

Lee Naish
lee@unimelb.edu.au

Abstract

Full disclosure of votes in STV elections can allow coercion of voters by the use of "signature attacks", but limiting disclosure can make independent verification of results impossible. We propose disclosure of a subset of the preferences in each vote, namely those that are actually used in the count. This scheme is easy to implement, permits verification of the tally, and combats signature attacks to a large degree.

Keywords: signature attack, STV, voter coercion, verification

1 Introduction

It has been noted that full disclosure of votes in STV elections with a reasonably large number of candidates provides a means for coercing voters. With proposed methods of limited disclosure it may be impossible to independently verify from the disclosed information that the votes have been counted correctly and/or verification may be impractically complex. Here we propose a new method of limited disclosure that combats coercion but allows independent verification of the tally and does not add significant complexity to the counting process. We first discuss the coercion method and two proposed solutions. We then present our method and discuss the difficulty of coercion if it is used.

2 The Italian attack

Otten [4] has noted the potential for coercion of voters in STV elections with a reasonably large number of candidates. This method of coercion is known as "the Italian attack" after its apparent use by the Mafia in Italy in the 1970s and 1980s. To elect members of the Italian parliament, voters would choose a party and had the option of expressing numeric preferences for several candidates within the list for that party, and lists of 40 or more candidates were not uncommon [4].

More technically, this form of coerced voting is known as a signature attack. A coercer tells each coerced voter precisely how to vote. Typically, a first preference for the coercer's preferred candidate is followed by some permutation of the other candidates that is very unlikely to be chosen by any other voter. This is the "signature" that, with high probability, uniquely identifies the voter. Each coerced voter is told a different permutation. The number of permutations of N candidates is N factorial, so with a reasonably large number of candidates it is easy enough to find many permutations that can identify coerced voters with a high degree of confidence. If votes are disclosed after the election, the coercer can reward or punish each coerced voter depending on whether or not a vote with their particular signature was cast.

It is possible to guard against this kind of attack by disclosing less information. However, it is desirable to disclose enough information to permit verification that the result of the tally is correct. For example, the Electoral Commission of the Australian Federal State of Victoria currently has a project underway for a computerised voting system, motivated by privacy concerns for visually impaired voters [2]. The use of computers makes the process less transparent, but by using cryptographic methods it is possible to create a completely verifiable system. Ideally, the result would be verifiable while coercion would not be facilitated. Although we do not discuss cryptography in detail here, the method proposed is designed with complete verifiability in mind.

3 Otten's proposal

Otten [4] suggests not disclosing some later preferences in order to prevent unique identification of a ballot after disclosure. He suggests that later preferences be removed until there are at least three copies of each reported permutation. For example, “if there is 1 vote of BCDEFGA, 1 of BCDEFAG and 1 of BCDEGAF then the fact that there were 3 votes of BCDExxx would be published”. Unfortunately, this proposal does not guarantee that the tally can be verified, as it is possible that the result depends on the later preferences of some of these ballots. Also, it is not clear that there will be a sufficiently large “crowd” to hide in. Choosing a larger number improves anonymity but decreases the chance of being able to verify the tally.

4 The Shuffle-sum proposal

Benaloh et al. [1] describe a scheme where votes are encrypted in such a way that each stage of the tally can be verified. For example, the fact that there were 100 first preferences for candidate A, say, would be revealed, but the other preferences of those ballots would not be revealed. When ballots were transferred, for example, when a candidate was excluded, they would be shuffled and re-encrypted. For example, if B and then A were excluded, the shuffling and re-encryption would make it impossible to distinguish between ballots of the form BAxxx and Axxxx. While this avoids signature attacks to the greatest possible extent, it is costly. With several dozen candidates, an encrypted ballot can take a megabyte or more of space. Verifying a tally requires the (re)encrypted version of each ballot at each stage of the count. Complex algorithms must be run on potentially many gigabytes of data. These practical considerations make it infeasible for the Victorian Electoral Commission to use the shuffle-sum proposal, despite some concern over signature attacks [2]

5 Properties of STV counting algorithms

Before moving on to our proposal, we discuss some key properties of STV counting algorithms that our proposal relies on. These properties hold for “traditional” STV counting rules that do not prescribe restarting after an exclusion. For rules that do prescribe a restarting after an exclusion, such as Meek and Warren, not all the properties hold. Our methods can be adapted to such rules, but more information will be disclosed, and hence signature attacks will be not as certainly prevented. Here we concentrate on traditional counting rules. There are four key properties of interest to us:

1) The counting procedure is sequential, punctuated by points where candidates are declared elected or excluded. No candidate is declared elected or excluded more than once, so the sequence of candidates declared elected or excluded defines a permutation of a subset of the candidates. It may be a strict subset of the candidates because some candidates may be excluded by default when the tally ends with the last candidate declared elected. We will call this permutation of a subset of the candidates, specified by the order in which candidates are either elected or excluded, the *tally sequence*. For example, with two vacancies, if B is excluded, D is elected, A excluded then E elected, the tally sequence would be BDAE.

2) After a candidate is declared elected or excluded, all preferences for that candidate on ballots are ignored in the counting process.

3) For each ballot paper, preferences are examined in order. It is possible that not all preferences, particularly later preferences, will be examined, and some earlier ones may be examined but ignored, due to 2). We will call the sequence of preferences examined and not ignored a ballot sequence. With the example tally in 1) above, the ballot ABCDE would have the ballot sequence AC. The first preference, A, is used and later when A has been excluded, B is ignored (since B has also been excluded at that point) and C is used. In a manual count, the ballot sequence is the path the ballot paper takes as it moves from candidate to candidate in the count. The ballot

DABCE could have the ballot sequence DAC (if the ballot was transferred as part of the surplus of D) or D (if D obtained exactly one quota of votes or if D had a surplus but the rules specified that only BDxxx ballots should be transferred, for example).

4) Each ballot sequence is a (not necessarily contiguous) subsequence of the tally sequence, possibly followed by a single candidate who is not in the tally sequence. If B appears before A in the tally sequence, for example, A cannot appear before B in any ballot sequence (if the ballot ever leaves candidate A, any preference for B will be ignored, since B will already have been declared elected or excluded). The last candidate on a ballot sequence may be a continuing candidate at the point the last candidate is declared elected, so they are not on the tally sequence. Other candidates in the ballot sequence must be in the tally sequence since the only trigger for moving a ballot to another (later) preference is when the candidate to whom the ballot is currently assigned is declared elected or excluded.

6 Our proposal

Our proposal is to disclose just the ballot sequence for each ballot.

The ballot sequences can be determined by very simple modifications to the counting method. We simply need a flag for each preference on each ballot. When the preference is used, it is flagged, and at the end of the count the ballot sequences can be output along with the successful candidates and tally sequence and/or detailed tally. For Meek and other more complex rules, the same method can be used; typically more preferences will be flagged and so ballot sequences will be longer.

Verification of the tally is straightforward—we can simply redo the count with the ballot sequences rather than the original ballots. This will result in an identical tally since the only difference in the two sets of ballots is that preferences which were never used have been removed.

7 Resilience against signature attacks

We now discuss how well our proposal guards against the standard form of the Italian attack and an alternative signature attack. However, we first make an observation about the maximum information content in ballot sequences.

Suppose we have N candidates and K are continuing candidates at the end of the tally. If complete ballots are disclosed, there are $N!$ possibilities for each one. If only ballot sequences are disclosed there are $(K + 1)2^{N - K}$ possibilities for each one. This is due to 4) above: The $K + 1$ factor comes from the choice of continuing candidates at the end of the ballot sequence (the +1 for the case where there is no continuing candidate at the end). The tally sequence has length $N – K$, and each candidate in the sequence may or may not be in the ballot sequence.

8 The standard Italian attack

Although $(K + 1)2^{N - K}$ is much less than $N!$, it is still likely to be large enough for sufficient unique “signatures” to be found. However, the possible ballot sequences depend on the tally sequence, which is only known after the tally has been computed. A coercer would have to be able to accurately predict the tally sequence in order to use this number of ballot sequences for an attack, and for each candidate whose stage of election/exclusion cannot be reliably predicted, the number of ballot sequences that can be used is halved.

Furthermore, the standard form of the Italian attack relies on the “signature” appearing in preferences after the choice of the coercer. Suppose the coercer wants candidate C elected. In the standard Italian attack, coerced voters would be told to mark C as their first preference. The signature could not contain any candidate declared elected or excluded before C, since those preferences would not appear in the ballot sequence. Furthermore, if C is the last candidate declared elected, no signature is revealed at all, and if C is the last candidate excluded, not enough information is revealed to identify significant numbers of voters.

In most situations, attempting to coerce voters is risky, and the greater the number of voters coerced, the greater the risk—there are severe consequences if you are caught. It is only worthwhile if the risk is outweighed by the increased chance of C being elected. In general, coercing significantly more voters than necessary is not a good strategy. For example, if C is already a popular candidate, likely to be elected, coercion is unwise. The situations favouring coercion as a strategy are precisely those where C is the last candidate declared elected (or excluded if the strategy doesn't quite work). These are exactly the situations where revealing only ballot sequences reveals minimal signature information. We thus conclude that our proposal should be very effective at guarding against the standard form of Italian attack.

9 Using early preferences as a signature

Ballot sequences for votes that elect candidate C can contain information encoded in the sequence of candidates elected or excluded before C. A coercer who had some knowledge of this sequence could thus use a form of signature attack. If the coercer had enough loyal supporters who would vote as instructed (without any coercion and the need for signatures on ballots), these supporters could influence the order of exclusion of several “dummy” candidates who stand at the behest of the coercer. These dummy candidates could be used for signatures in votes which eventually deliver a preference to candidate C. We now briefly analyse such an attack. We assume the best case scenario for the coercer, where only their loyal supporters and coerced voters vote for the dummy candidates.

With N dummy candidates, $D_1, . . . , D_N$, up to $2^N - 1$ signatures can be encoded. Assuming they are excluded in that order, 2^{N-1} of the coerced votes will have D_1 as the first preference, 2^{N-2} will have D_2 as the first preference, and so on. To ensure D_1 is excluded first, there must therefore be at least $2^{N-1} – 2^{N-2}$ loyal supporters with a first preference for D_2, $2^{N-1} – 2^{N-3}$ loyal supporters with a first preference for D_3, and so on. Thus, approximately $(N – 2)2^{N-1}$ votes from loyal supporters are required to ensure D_1 is excluded first.

Furthermore, when D_1 is excluded, half the preferences from coerced votes will go to D_2, a quarter to D_3 and so on. Thus the totals for the coerced votes at this stage of the count will be 2^{N-1} for D_2, 2^{N-2} for D_3, and so on. Similarly, after D_2 is excluded, D_3 will have 2^{N-1} coerced votes and D_4 will have 2^{N-2} coerced votes. The votes from loyal supporters must be sufficient to ensure the correct exclusion at each stage. It would be sufficient to have $2^{N-2} + 1$ first preferences for D_2, $2(2^{N-2} + 1)$ first preferences for D_3, $3(2^{N-2} + 1)$ first preferences for D_4, and so on, a total of about $2^{N-3}N^2$ votes from loyal supporters. Somewhat fewer than this number is sufficient in theory, since the first preference votes for D_2 can be re-used after D_2 is excluded, to help top up the totals of D_4 etc. (so that D_3 is excluded next). However, additional votes are advisable to combat the possibility that some voters who are neither loyal nor coerced may cast votes for the dummy candidates. In addition, the number of coerced voters must be somewhat less than 2^N to account for the permutations of preferences used by the loyal supporters.

Although we have not established the precise optimal relationship between the number of dummy candidates, the number of loyal supporters and the number of coerced votes, it seems this strategy may be plausible for very small elections, but is unlikely to be successful for larger elections that have careful oversight from electoral authorities. For example, to obtain an extra 1,000 votes through coercion, about 12,000 loyal supporters must be instructed to vote in the right ways and 10 dummy candidates must stand.

10 Conclusions

To verify the correctness of a tally in STV elections, some voting information must be disclosed. If all information is disclosed, signature attacks such as the Italian attack can be used to enable coercion of voters. Previous work proposed a cryptographic method that minimises the information disclosed and allows verification in theory but makes it difficult in practice due to the size of the data produced and the complexity of the algorithms used. This paper proposes an alternative scheme for less than full disclosure. It entails disclosure of more information than the cryptographic method but is much simpler to implement and makes verification much easier. It combats the standard Italian attack effectively. There are other possible attacks that involve standing dummy candidates and using the votes of loyal supporters to attempt to ensure that these candidates are excluded in a particular order. However, particularly for larger scale elections, our proposal seems to provide a reasonable compromise between the ease of verifying the correctness of the tally and the risk of signature attacks being used to coerce voters.

11 References

[1] Benaloh, J., Moran, T., Naish, L., Ramchen, K., and Teague, V. (2009). Shuffle-sum: Coercion-resistant verifiable tallying for STV voting, *IEEE Transactions on Information Forensics and Security* 4 (4).

[2] Burton, Craig et al. (2012). Using Prêt à Voter in Victorian State elections, https://www.vec.vic.gov.au/files/RP-EVT.pdf (viewed 11/2012).

[3] Di Cosmo, Roberto, On privacy and anonymity in electronic and non electronic voting: the ballot-as-signature attack. 2007 http://hal.archives-ouvertes.fr/docs/00/15/87/07/PDF/PrivacyTests.pdf (viewed 10/2012).

[4] Otten, J. (2003). Fuller disclosure than intended, *Voting matters* 17, p. 8.

About the Author

Lee Naish is a senior lecturer in the Department of Computing and Information Systems, The University of Melbourne. His main research areas are programming languages and debugging, but he has also published in the area of voting systems. He is a life member of the Proportional Representation Society of Australia (Victoria-Tasmania) inc. and a long-standing member of its council. His more colourful pursuits have included rock climbing, unicycling and fire breathing.