# First-past-the-post suffices for ranked voting

Ben Smyth

Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg

September 11, 2017
(Revised September 19, 2018)

**Abstract**

This manuscript introduces a technique that enables first-past-the-post voting systems to be used for ranked voting and applies the technique to the Helios voting system.

## 1 Introduction

Smyth, Frink & Clarkson [SFC17, §2] investigate the class of voting systems that consist of the following four steps. First, a tallier generates a key pair and (optionally) a registrar generates credentials for voters. Secondly, each voter constructs and casts a ballot for their preferred candidate. Thirdly, the tallier tallies the ballots and announces a frequency distribution of candidate preferences. Finally, voters and other interested parties check that the distribution corresponds to preferences expressed in ballots. This class includes first-past-the-post voting systems.

**Contribution.** This manuscript introduces a technique that enables first-past-the-post voting systems to be used for ranked voting (§2) and applies the technique to the Helios voting system (§3).

## 2 Ranked voting from first-past-the-post

Rankings can be represented as candidates and voters can cast ballots for the candidate that represents their ranking.[1] Such ballots can be tallied to derive the frequency distribution of rankings and the winner can be determined

[1] The idea of using candidates to represent more complicated data structures is not new. Indeed, candidates have been used to represent prices in auction schemes constructed from voting systems [MSQ14, QS18].

from this distribution. For instance, rankings $A{>}B{>}C$, $A{>}C{>}B$, $B{>}A{>}C$, $B{>}C{>}A$, $C{>}A{>}B$ and $C{>}B{>}A$ can be represented as candidates 1–6. Moreover, given two ballots for candidate 1, one ballot for candidate 4 and one ballot for candidate 6, tallying produces frequency distribution $A{>}B{>}C$:2, $A{>}C{>}B$:0, $B{>}A{>}C$:0, $B{>}C{>}A$:1, $C{>}A{>}B$:0, $C{>}B{>}A$:1, from which candidate $A$ can be determined as the winner.

Rankings can always be mapped to candidates using this technique. However, the technique cannot be applied to all voting systems, because voting systems may bound the number of candidates. For instance, a voting system for referendums may implicitly bound the number of candidates to two, hence, that voting system cannot be used to rank three candidates, since the six possible rankings cannot be mapped to two candidates.

# 3 Case study: Helios

Applicability of the technique, and complexity implications, can be shown by applying it to the Helios voting system in each of its two tallying modes.

## 3.1 Tallying by homomorphic combinations

Helios with tallying by homomorphic combination of ciphertexts [AMPQ09] works as follows.

1. The tallier generates a key pair for an asymmetric additively-homomorphic encryption scheme, proves correct key generation in zero-knowledge, and outputs the public key coupled with the proof.

2. A voter selects their preferred candidate $v$ from a list of candidates $1, \dots, \ell$ and computes ciphertexts $\mathsf{Enc}(pk, m_1), \dots, \mathsf{Enc}(pk, m_{\ell-1})$ such that if $v < \ell$, then plaintext $m_v$ is 1 and the remaining plaintexts are all 0, otherwise, all plaintexts are 0. The voter also computes proofs $\sigma_1, \dots, \sigma_\ell$ so that this can be verified. The voter casts the ciphertexts and proofs as their ballot.

3. The tallier forms a matrix of ciphertexts from ballots for which all proofs hold, homomorphically combines the ciphertexts in each column to derive the encrypted frequency distribution of candidate preferences, decrypts those homomorphic combinations to reveal the frequencies of candidates $1, \dots, \ell - 1$, computes the frequency of candidate $\ell$ by subtracting the frequencies of other candidates from the number of rows in the matrix, and announces the frequencies, along with proofs demonstrating correct decryption.

4. Voters and other interested parties recompute the homomorphic combination, check proofs, and accept the frequency distribution if these checks succeed.

Hence, in the context of the example given in Section 2, the technique leads to the following matrix of ciphertexts:

$$\begin{array}{lllll}
\mathsf{Enc}(pk,1) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) \\
\mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) \\
\mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,1) & \mathsf{Enc}(pk,0) \\
\mathsf{Enc}(pk,1) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0)
\end{array}$$

These ciphertexts will be homomorphically combined to derive ciphertexts

$$\begin{array}{lllll}
\mathsf{Enc}(pk,2) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,0) & \mathsf{Enc}(pk,1) & \mathsf{Enc}(pk,0)
\end{array}$$

and decrypted to reveal frequencies

$$A{>}B{>}C{:}2 \quad A{>}C{>}B{:}0 \quad B{>}A{>}C{:}0 \quad B{>}C{>}A{:}1 \quad C{>}A{>}B{:}0,$$

which permit the deduction of frequency $C{>}B{>}A{:}4-2-1$. This shows that the technique can be applied to derive a ranked voting system. However, ballots include $\ell! - 1$ ciphertexts and $\ell$ proofs, where $\ell$ is the number of candidates. Moreover, tallying requires $(n-1)\cdot(\ell!-1)$ homomorphic combinations and $\ell!-1$ proofs, where $n$ is the number of ballots with valid proofs. Such complexity is avoided when applying the technique to Helios's other tallying mode.

### 3.2 Tallying by mixnet

Helios with tallying by mixnet [Adi08, BGP11, Smy18b] works as follows.

1. As per the previous tallying mode, without requiring that the asymmetric homomorphic encryption scheme is additively-homomorphic.
2. A voter encrypts their preferred candidate, proves correct ciphertext construction, and casts the ciphertext coupled with the proof as their ballot.
3. The tallier discards any ballots for which proofs do not hold, mixes the ciphertexts in the remaining ballots, decrypts the ciphertexts output by the mix to reveal the frequency distribution of candidate preferences, and announces that distribution, along with proofs demonstrating correct decryption.
4. Voters and other interested parties check the proofs and accept the frequency distribution if these checks succeed.

Hence, in the context of the example given in Section 2, the technique results in the following:

$$\begin{array}{lcll}
\mathsf{Enc}(pk,1) & & \mathsf{Enc}(pk,6) & C{>}B{>}A \\
\mathsf{Enc}(pk,6) & \text{Mix} & \mathsf{Enc}(pk,1) & A{>}B{>}C \\
\mathsf{Enc}(pk,4) & & \mathsf{Enc}(pk,1) & A{>}B{>}C \\
\mathsf{Enc}(pk,1) & & \mathsf{Enc}(pk,4) & B{>}C{>}A
\end{array}$$

This permits the frequency distribution of candidate preferences to be derived. Again showing that the technique can be applied to derive a ranked voting system. Moreover, the complexity of the previous application is avoided. Indeed, ballots include just a single ciphertext and a single proof.

# 4 Security considerations

To enable first-past-the-post voting systems to be used for ranked voting, we represent rankings as candidates. No changes to the underlying voting system are required. Hence, ranked voting can be achieved with the same security as the underlying system. That is, any security statement that holds for all candidates, also holds for rankings, because rankings are represented as candidates and security statements must hold for all candidates. It follows, for instance, that a voting system satisfying the definition of ballot secrecy by Smyth [Smy18a], assures that a voter's ranking is not revealed to anyone (since candidates cannot be revealed). Moreover, a voting system satisfying definitions of universal and individual verifiability by Smyth, Frink & Clarkson [SFC17], assures that anyone can check whether an outcome corresponds to rankings expressed in collected ballots (since checks can be performed for candidates) and a voter can check whether their ballot is collected (since the statement is independent of candidates and rankings). Nonetheless, security guarantees beyond what can be derived from the underlying system might be desired. For example, it might be desired that relative rankings between losing candidates are not revealed, which is not possible in our context.

# 5 Related work

Clarkson, Chong & Myers [CCM07, §8] propose an encoding of rankings as matrices such that cell $i, j$ is 1 if candidate $i$ is favoured over candidate $j$ and 0 otherwise. (Ties between candidates are expressed when cells $i, j$ and $j, i$ are equal.[2]) This encoding allows intransitive preferences. E.g., a voter might express a preference for $A$ over $B$, $B$ over $C$, and $C$ over $A$. And an additional mechanism is required to avoid such intransitivity (zero-knowledge proofs are suggested). Clarkson, Chong & Myers explain how their encoding could be used to enable ranked voting with Civitas in a manner that avoids covert channels.

# 6 Conclusion

This manuscript introduces a technique that enables first-past-the-post voting systems to be used for ranked voting. The technique involves representing rankings as candidates. Hence, the underlying voting system does not require

[2] A variant of the encoding could disregard cells $j, i$ such that $j < i$ to avoid ties being expressed.

redesign. Thereby permitting ranked voting with the same security as the underlying system.

## References

[Adi08] Ben Adida. Helios: Web-based Open-Audit Voting. In *USENIX Security'08: 17th USENIX Security Symposium*, pages 335–348. USENIX Association, 2008.

[AMPQ09] Ben Adida, Olivier de Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios. In *EVT/WOTE'09: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. USENIX Association, 2009.

[BGP11] Philippe Bulens, Damien Giry, and Olivier Pereira. Running Mixnet-Based Elections with Helios. In *EVT/WOTE'11: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. USENIX Association, 2011.

[CCM07] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a Secure Voting System. Technical Report 2007-2081, Cornell University, 2007.

[MSQ14] Adam McCarthy, Ben Smyth, and Elizabeth A. Quaglia. Hawk and Aucitas: e-auction schemes from the Helios and Civitas e-voting schemes. In *FC'14: 18th International Conference on Financial Cryptography and Data Security*, volume 8437 of *LNCS*, pages 51–63. Springer, 2014.

[QS18] Elizabeth A Quaglia and Ben Smyth. Secret, verifiable auctions from elections. *Theoretical Computer Science*, 730:44–92, 2018.

[SFC17] Ben Smyth, Steven Frink, and Michael R. Clarkson. Election Verifiability: Cryptographic Definitions and an Analysis of Helios and JCJ. Cryptology ePrint Archive, Report 2015/233, 2017.

[Smy18a] Ben Smyth. Ballot secrecy: Security definition, sufficient conditions, and analysis of Helios. Cryptology ePrint Archive, Report 2015/942, 2018.

[Smy18b] Ben Smyth. Verifiability of Helios Mixnet. In *Voting'18: 3rd Workshop on Advances in Secure Electronic Voting*, LNCS. Springer, 2018.