

Exploiting re-voting in the Helios election system

Maxime Meyer^a, Ben Smyth^b

^a*Vade Secure Technology Inc., Montreal, Canada*

^b*Interdisciplinary Centre for Security, Reliability and Trust,
University of Luxembourg, Luxembourg*

Abstract

Election systems must ensure that representatives are chosen by voters. Moreover, each voter should have equal influence. Traditionally, this has been achieved by permitting voters to cast at most one ballot. More recently, this has been achieved by tallying the last ballot cast by each voter. We show this is not achieved by the Helios election system, because an adversary can cause a ballot other than a voter’s last to be tallied. Moreover, we show how the adversary can choose the contents of such a ballot, thus the adversary can unduly influence the selection of representatives.

1. Introduction

An election is a decision-making procedure to choose representatives [1, 2, 3, 4]. Choices are made by voters, and this must be ensured by election systems, as prescribed by the United Nations [5, Article 21], the Organization for Security and Cooperation in Europe [6, Paragraph 7.3], and the Organization of American States [7, Article 23]. These organisations also prescribe that systems must ensure that voters have equal influence in the decision [5, 6, 7]. This has led to the emergence of the following eligibility and non-reusability requirements [8, 9].¹

- Eligibility. Choices are only made by voters.

¹Kremer & Ryan capture both requirements in a single, informal definition, namely, “only...voters can vote, and only once” [8], whereas Backes *et al.* decouple that definition into “only...voters can vote” and “every voter can vote only once” [9]. (We refer to voters and non-voters, whereas Kremer & Ryan and Backes *et al.* distinguish non-voters from ‘legitimate voters’ and ‘eligible voters.’)

Eligibility ensures that non-voters cannot (directly) influence the decision. For instance, national elections typically require that voters are citizens of the nation, thus, eligibility forbids influence from foreign citizens.²

- Non-reusability. Only one choice of each voter has influence.

Non-reusability ensures that each voter can contribute at most one choice, hence, voters have equal influence. In addition, for verifiable elections [12, 13, 14, 15, 16], non-reusability is useful to aid recovery from failure (since voters can “*vote, verify, and revote until verification succeeds*” [17, §1]).

Election systems have traditionally permitted each voter to cast at most one choice. More recent systems permit multiple choices (e.g., [18, 19, 20, 21, 22, 23]) and a voter’s last choice should have influence. We strengthen the aforementioned non-reusability requirement to capture such influence.

- Strong non-reusability. Only the last choice of each voter has influence.

Strong non-reusability enables voters to change their choices, which provides flexibility, and aids education (since voters can “*ask the help of anyone for submitting a random ballot, and then re-voting privately afterwards*” [20, §3.3]). By comparison, (weak) non-reusability does not enable voters to change their choices, because that notion does not specify which ballot should have influence. Hence, it is permissible for a choice, other than the voter’s last, to have influence. Consequently, voters cannot change their choices, because voters do not know which of their choices will have influence. Thus, the notions of non-reusability by Kremer & Ryan [8] and Backes *et al.* [9] are too weak to analyse an interesting property of recent election systems; a slight strengthening of their notions is necessary.

Eligibility and non-reusability are fundamental requirements of election systems [5, 6, 7], as-is strong non-reusability when voters are permitted to change their choices. These requirements all assume that the adversary’s capabilities are limited to controlling the communication channel and that the election system is operated in the prescribed manner, hence, they are not intended to exclude attacks that arise when the election system is subverted by the adversary (to authenticate non-voter ballots, for instance). By

²We concede that non-voters may indirectly influence the decision, e.g., voters may be swayed by disinformation [10, 11].

comparison, verifiability requirements assume the election system has been subverted and are intended to enable the detection, rather than exclusion, of attacks [12, 13, 14, 15, 16]. It follows that an election system that satisfies eligibility, non-reusability, and strong non-reusability is invulnerable to attacks against those requirements when the election system is operated in the prescribed manner, whereas a verifiable system might be vulnerable to attacks, but those attacks can be detected. Thus, eligibility, non-reusability, and strong non-reusability should be satisfied regardless of whether verifiability is, because election systems operated in the prescribed manner should prevent attacks by network adversaries, rather than just enabling attack detection.

We analyse Helios [20]: an open-source, web-based election system,³ which has been used by the International Association of Cryptologic Research (IACR), the ACM, the Catholic University of Louvain, and Princeton University [24]. Helios uses a third party to authenticate voters' ballots, which suffices for eligibility, assuming the third party is trusted. Authenticated ballots are listed alongside voter identities and at most one ballot is listed alongside each identity, which suffices for non-reusability. Any other ballots are archived. Auditing can be used to statistically determine whether non-voter ballots are incorrectly authenticated by an untrusted third party or whether unauthenticated ballots are listed. For instance, voters can be asked whether the ballot alongside their identity is theirs, to determine if the ballot was incorrectly authenticated or simply unauthenticated. Albeit cooperation and honesty of some voters is required for auditing, and sufficiently many malicious voters can manipulate audits.

Contribution. We show that the archiving mechanism used by Helios is insufficient to ensure strong non-reusability, in the presence of an adversary that is able to delay messages sent on the network. In particular, the adversary can cause a choice other than a voter's last to be counted. Moreover, we show how the adversary can pick the choice, in a poll station with a malicious election supervisor. Although malice can be detected by voters that perform verifiability checks once voting closes, recovery is only possible before tallying commences.

³<https://vote.heliosvoting.org>, accessed 11 Aug 2017.

2. Analysis of Helios

2.1. Protocol description

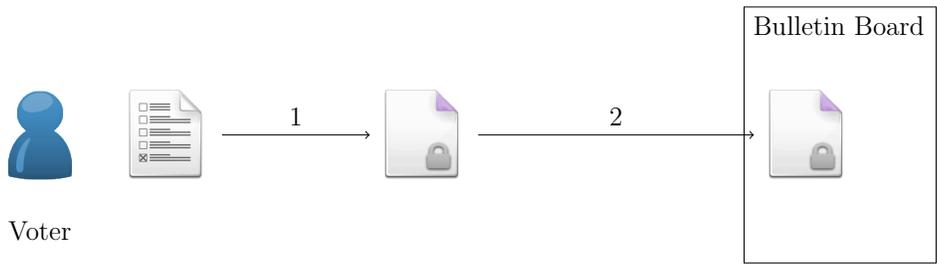
An execution of Helios (Figure 1) proceeds as follows. First, a voter casts a ballot for their choice: the voter encrypts their choice (1) and sends their encrypted choice to the bulletin board (2). Secondly, the voter authenticates their encrypted choice to the bulletin board, to prove they are indeed a voter. The authentication process uses OAuth [25],⁴ which is reliant on a third party. The process proceeds as follows. The voter authenticates to a third party (3), the third party generates a token for the voter (4), the voter sends the token to the bulletin board (5), and the bulletin board relays the token to the third party (6). The third party checks whether the token is valid and notifies the bulletin board of the token’s validity (7). If the token is valid, then the bulletin board accepts the voter’s encrypted choice. Hence, the bulletin board contains the voter’s authenticated encrypted choice. In addition, the bulletin board archives any encrypted choice previously accepted for that voter,⁵ which is intended to ensure that only the last choice of each voter has influence. Finally, the bulletin board homomorphically combines the accepted encrypted choices (8), the administrator decrypts the homomorphic combination (9), and the bulletin board reveals those decrypted choices (10). Helios satisfies eligibility, because encrypted choices are only accepted by the bulletin board when accompanied by a token authenticating the voter that constructed the encrypted choice. Moreover, Helios satisfies non-reusability too, because, upon acceptance, the bulletin board archives any encrypted choice previously accepted for that voter. But, this is insufficient for strong non-reusability.

2.2. Vulnerability

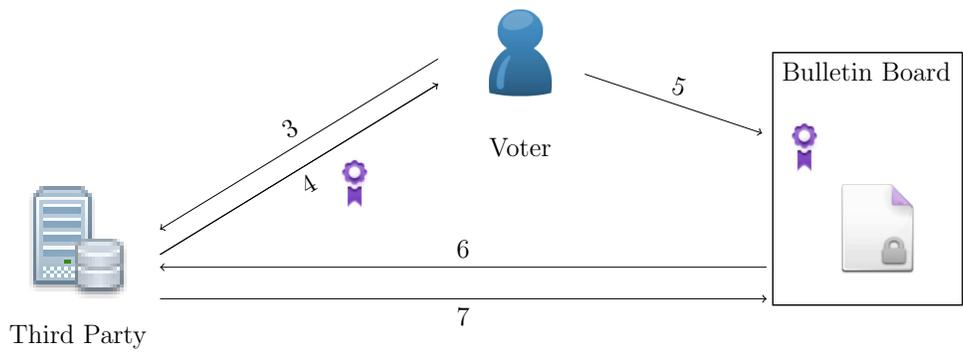
The flow of our exploit initially corresponds to an honest execution: a voter casts a ballot for their choice, as per Figure 1a. The remaining steps (Figure 2) proceed as follows. First, the adversary intercepts a voter’s token: the voter authenticates to a third party (1), receives an authentication token (2), and sends the token to the bulletin board (3), but it is intercepted by

⁴Other authentication methods are also supported.

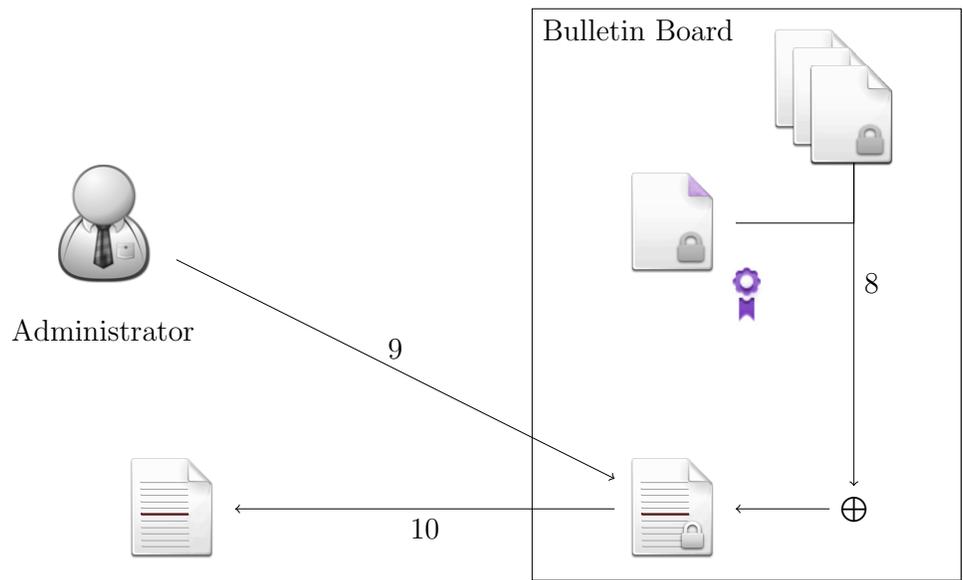
⁵See `Voter.last_cast_vote()` in <https://github.com/benadida/helios-server/blob/9fa74a2bef41c0c344f1c9a6f1c28a36f93347ea/helios/models.py>, accessed 11 Aug 2017.



(a) Casting a ballot



(b) Authenticating the ballot



(c) Tallying

Figure 1: Helios protocol flow

the adversary (4).⁶ (We indicate the ballot-token relation by colouring the top right-hand corner of the ballot and the token in purple.) Thus, the bulletin board contains an unauthenticated encrypted choice and is awaiting an authentication token for that encrypted choice. Next, the adversary waits until the voter casts another encrypted choice (5), authenticates with the third party (6), receives a token (7), and sends the token to the bulletin board (8). (We indicate the ballot-token relation using green colouring.) Thus, the bulletin board can authenticate the voter’s second ballot. Finally, the adversary releases the intercepted token and it is received by the bulletin board (9). Thus, the bulletin board will accept the voter’s first ballot, and archive the voter’s second ballot (10). Consequently, the voter’s first choice is counted, rather than their second. Hence, strong non-reusability is not satisfied, because only the last choice of the voter should have influence, which is not the case.

Video demonstration. The exploit is demonstrated in a supporting video [26].

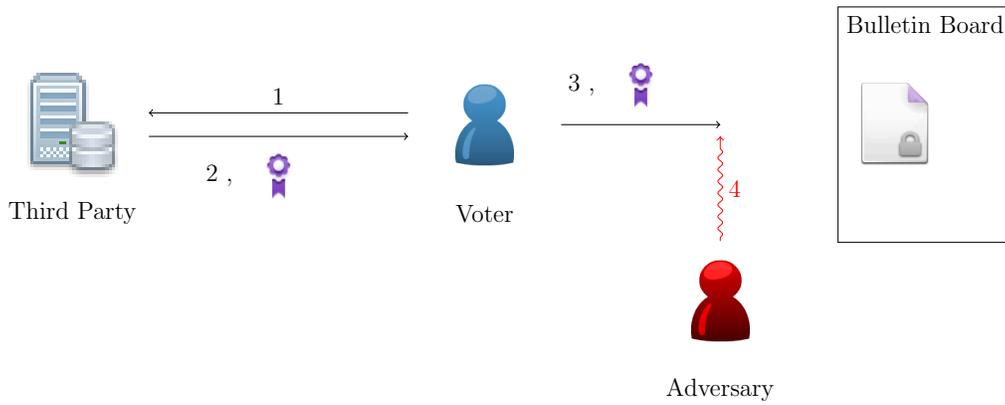
Helios developers Ben Adida and Olivier Pereira acknowledge the existence of this vulnerability, but they contend it would be detected.⁷ We will discuss detection mechanisms in the following section.

2.3. Impact

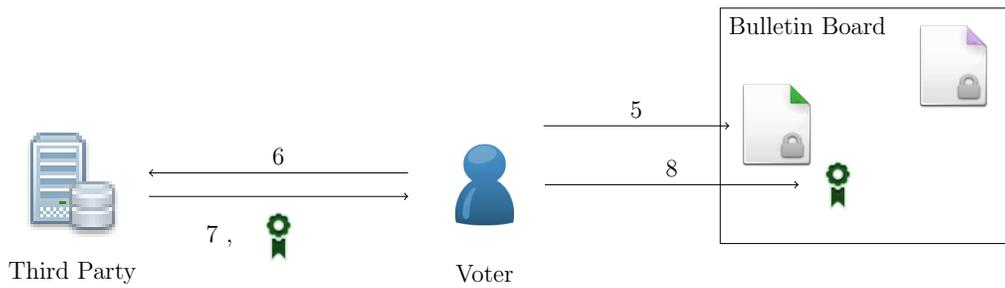
Let us now consider the possibility of an adversary unduly influencing an election’s outcome, in settings where Helios is deployed in voting terminals located at poll stations. In such settings, a malicious election supervisor could offer to demonstrate the Helios system to a voter, under the guise of education. During the demonstration, the supervisor could suggest that the voter selects a particular choice. This should not cause suspicion, because Helios is intended to permit voters to change their choices (§1). Once the voter casts the demonstration ballot, it could be intercepted, perhaps by a router in the

⁶An adversary can intercept packets even when they are encrypted. For example, packets sent over a TLS connection, i.e., encrypted packets, can be intercepted. Intercepting a TLS packet prevents further data from being received on *that* TLS connection (until the packet is released), but data may be received on other TLS connections (of which there are many), because TLS does not guarantee ordering of messages between connections. (Multiple TLS connections are maintained to reduce latency.) Hence, TLS does not prevent further communication between the voter and the bulletin board.

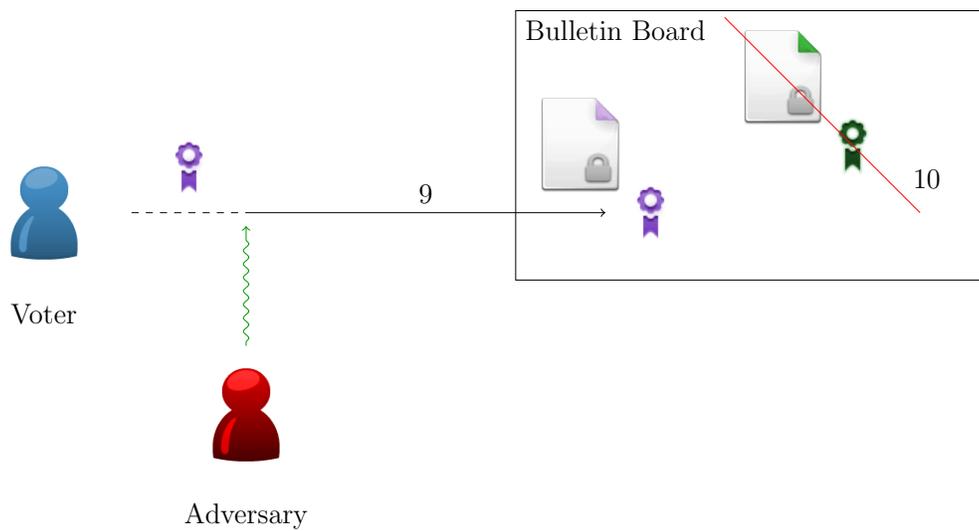
⁷Email communication, April 2014.



(a) Token interception



(b) Casting a second ballot



(c) Release intercepted token

Figure 2: Helios exploit flow

polling station that the supervisor controls. After the demonstration, the supervisor could instruct the voter to re-vote in private. Once the voter leaves the poll station, the intercepted ballot could be released. Consequently, the supervisor’s choice is tallied, rather than the voter’s, thereby demonstrating the possibility of undue influence.

We acknowledge that the voter can discover that malice has taken place, because Helios satisfies individual verifiability [15, 13] and the voter can check whether the bulletin board accepted their second encrypted choice. However, it is well-known that many voters do not perform checks necessary for verifiability and voting systems rely on checks being performed by sufficiently many diligent voters [27, §2.1.6]. Thus, the exploit is particularly effective against voters that do not perform checks. Moreover, even if malice is detected, recovery is only possible when a voter successfully casts another encrypted choice (before tallying), hence, the exploit can also be effective against voters that detect malice. Effectiveness can be improved by releasing the intercepted ballot just before voting closes, this not only reduces the voter’s opportunity to detect malice, but also forces the voter to convince officials that they should be able to cast another ballot after voting closes, which is problematic, because there is no convincing evidence that any malpractice has taken place. Once tallying commences, the voter cannot recover, furthermore, given the absence of any evidence, victims have little recourse.

We believe that verification checks should serve as a last line of defence and that election systems should prevent many attacks, rather than merely being able to detect them (especially as detection does not imply the ability to recover). Hence, we believe eligibility and non-reusability are worthy of study independently of verifiability.

2.4. *Fixes*

We can patch the vulnerability by checking authentication token timestamps, timestamping ballots, coupling encrypted choices with counters, or proving knowledge of earlier encrypted choices *à la* Clarkson, Chong & Myers [28, §3.3]. We favour solutions using timestamps, since the other approaches require the voter to maintain state. Moreover, timestamps have been acknowledged as a possibility for a fix by Helios developer Olivier Pereira.⁸ We concede that timestamps increase the attack surface, since

⁸Email communication, April 2014.

an adversary may tamper with clocks. But, we stress that the third party is already assumed to construct authentication tokens correctly and that voters are already assumed to construct ballots correctly (or, at least, audit ballots to increase confidence of correct construction),⁹ hence, tampering with clocks might be precluded by those assumptions.

OAuth tokens may contain timestamps [29, §2.2] and these can be used to determine the order in which ballots were authenticated. Similarly, ballots could be extended to include timestamps which can be used to determine the order in which ballots were constructed.¹⁰ These timestamps can be used by the bulletin board to patch the vulnerability. Indeed, rather than archiving any encrypted choice previously accepted for a voter, the bulletin board can archive any encrypted choice associated with an earlier timestamp.

The validity of tokens can only be checked by the bulletin board, because tokens must remain secret. Thus, the bulletin board might convince itself that ballots are authenticated, but it cannot convince other parties. Developing an authentication mechanism that permits anyone to check the validity of authentication tokens, rather than just the bulletin board, would be an interesting direction for future work. Alternatively, voters can be issued with credentials and cryptography can be used to ensure that only voters can construct authorised ballots (i.e., authorised ballots are unforgeable [30, §1]). For instance, Quaglia & Smyth [31] replace tokens with digital signatures. But, solutions reliant on cryptography seem to require expensive infrastructures for voter credentials and seemingly ignore the problem of corruption during the registration procedure [15, §4]. Indeed, auditing is required to check whether any non-voters are issued credentials. Eliminating such audits is desirable, but perhaps impossible.

⁹Auditing ballots provides assurance that ballots (constructed by untrusted systems) are cast as intended, in particular, ballots encapsulate voters' choices. This property is complimentary to individual verifiability, which allows voters to check whether their ballot is accepted by the bulletin board.

¹⁰An encrypted choice comprises of El Gamal ciphertexts and non-interactive zero-knowledge proofs, and timestamps could be included in hashes used by proofs. This ensures that timestamps cannot be modified, if they are to be accepted by the bulletin board, because the board checks validity of proofs before accepting them.

3. Related work

Smyth & Pironti [32] identify a flaw in Helios’s sign-out procedure which can be exploited by TLS truncation attacks to dupe voters into believing they have successfully signed-out, when they have not. Thus, an adversary can make a choice on the voter’s behalf from the terminal used by the voter, thereby violating eligibility. Beyond eligibility and non-reusability, malleability has been exploited to launch violate ballot secrecy [33, 34, 35, 36, 37] and verifiability [15], and unsound proofs of knowledge have been exploited to violate verifiability [38].

4. Conclusion

We have shown that Helios does not satisfy strong non-reusability, because an adversary can cause a ballot other than a voter’s last to be tallied. In particular, the adversary can intercept the authorisation token associated with the ballot that the adversary wants tallied, wait until the voter has casts their last ballot, and then release the intercepted token. The released token causes the bulletin board to accept the ballot that the adversary wants tallied, and to archive the voter’s last ballot. Thus, strong non-reusability is not satisfied. We have also shown that an adversary can choose the contents of such ballots. In particular, the adversary can exploit the educational needs of voters to cast a ballot for the adversary’s choice, and cause that ballot to be tallied rather than the voter’s last, as we have explained. Consequently, adversaries can unduly influence the selection of representatives. Although victims may detect malice, there is no evidence that malpractice has taken place, hence, victims have little recourse. The vulnerability is due to the manner in which Helios interacts with OAuth. Hence, our exploit should generalise to other systems that use OAuth in a similar manner and to systems that use similar authentication mechanisms. We hope that this article leads to improvements in the Helios election system, advances understanding of authentication mechanisms, and helps system developers to integrate authentication mechanisms securely.

Acknowledgements

We thank Elizabeth Quaglia, Susan Thomson and our anonymous reviewers for feedback that helped improve this paper. Smyth’s work was partly

performed at INRIA, with support from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013) / ERC project *CRYSP* (259639).

References

- [1] A. Lijphart, B. Grofman, *Choosing an electoral system: Issues and Alternatives*, Praeger, 1984.
- [2] T. Saalfeld, *On Dogs and Whips: Recorded Votes*, in: *Parliaments and Majority Rule in Western Europe*, St. Martin’s Press, 1995, Ch. 16.
- [3] A. Gumbel, *Steal This Vote: Dirty Elections and the Rotten History of Democracy in America*, Nation Books, 2005.
- [4] R. M. Alvarez, T. E. Hall, *Electronic Elections: The Perils and Promises of Digital Democracy*, Princeton University Press, 2010.
- [5] *Universal Declaration of Human Rights (1948)*.
- [6] *Document of the Copenhagen Meeting of the Conference on the Human Dimension of the CSCE (1990)*.
- [7] *American Convention on Human Rights, “Pact of San Jose, Costa Rica” (1969)*.
- [8] S. Kremer, M. D. Ryan, *Analysis of an Electronic Voting Protocol in the Applied Pi Calculus*, in: *ESOP’05: 14th European Symposium on Programming*, Vol. 3444 of LNCS, Springer, 2005, pp. 186–200.
- [9] M. Backes, C. Hrițcu, M. Maffei, *Automated Verification of Remote Electronic Voting Protocols in the Applied Pi-calculus*, in: *CSF’08: 21st Computer Security Foundations Symposium*, IEEE, 2008, pp. 195–209.
- [10] *Putin’s Asymmetric Assault on Democracy in Russia and Europe: Implications for U.S. National Security (2018)*.
- [11] *Disinformation and ‘fake news’: Interim Report (2018)*.
- [12] J. D. Cohen, M. J. Fischer, *A Robust and Verifiable Cryptographically Secure Election Scheme*, in: *FOCS’85*, IEEE, 1985, pp. 372–382.

- [13] S. Kremer, M. D. Ryan, B. Smyth, Election verifiability in electronic voting protocols, in: ESORICS'10, Vol. 6345 of LNCS, Springer, 2010, pp. 389–404.
- [14] R. Küsters, T. Truderung, A. Vogt, Verifiability, Privacy, and Coercion-Resistance: New Insights from a Case Study, in: S&P'11, IEEE, 2011, pp. 538–553.
- [15] B. Smyth, S. Frink, M. R. Clarkson, Election Verifiability: Cryptographic Definitions and an Analysis of Helios and JCJ, Technical Report 2015/233, Cryptology ePrint Archive (2015).
- [16] A. Kiayias, T. Zacharias, B. Zhang, End-to-End Verifiable Elections in the Standard Model, in: EUROCRYPT'15, Vol. 9057 of LNCS, Springer, 2015, pp. 468–498.
- [17] B. Adida, C. A. Neff, Ballot casting assurance, in: EVT'06, USENIX, 2006.
- [18] A. Juels, D. Catalano, M. Jakobsson, Coercion-Resistant Electronic Elections, Cryptology ePrint Archive, Report 2002/165 (2002).
- [19] E. Maaten, Towards remote e-voting: Estonian case, *Electronic Voting in Europe-Technology, Law, Politics and Society* 47 (2004) 83–100.
- [20] B. Adida, O. Marneffe, O. Pereira, J. Quisquater, Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios, in: EVT/WOTE'09, USENIX, 2009.
- [21] K. Gjøsteen, *The Norwegian Internet Voting Protocol*, Springer, 2012, pp. 1–18.
- [22] V. Cortier, D. Galindo, S. Glondu, M. Izabachène, Election Verifiability for Helios under Weaker Trust Assumptions, in: ESORICS'14: 19th European Symposium on Research in Computer Security, Vol. 8713 of LNCS, Springer, 2014, pp. 327–344.
- [23] G. V. Post, Using re-voting to reduce the threat of coercion in elections, *Electronic Government, an International Journal* 7 (2) (2010) 168–182.
- [24] O. Pereira, *Internet Voting with Helios*, in: *Real-World Electronic Voting: Design, Analysis and Deployment*, CRC, 2016, Ch. 11.

- [25] D. Hardt, The oauth 2.0 authorization framework, RFC 6749, Internet Engineering Task Force (2012).
- [26] B. Smyth, S. Thomson, Helios Re-voting Attack, YouTube video, linked from <https://bensmyth.com/publications/2016-attacking-eligibility-in-Helios/> (2014).
- [27] M. Bernhard, J. Benaloh, J. A. Halderman, R. L. Rivest, P. Y. A. Ryan, P. B. Stark, V. Teague, P. L. Vora, D. S. Wallach, Public evidence from secret ballots, in: E-Vote-ID'17: 10th International Conference for Electronic Voting, LNCS, Springer, 2017, pp. 84–109.
- [28] M. R. Clarkson, S. Chong, A. C. Myers, Civitas: Toward a Secure Voting System, in: S&P'08, IEEE, 2008, pp. 354–368.
- [29] E. J. Richer, Oauth 2.0 token introspection, RFC 7662, Internet Engineering Task Force (2015).
- [30] B. Smyth, A foundation for secret, verifiable elections, Cryptology ePrint Archive, Report 2018/225 (2018).
- [31] E. A. Quaglia, B. Smyth, Authentication with weaker trust assumptions for voting systems, in: AFRICACRYPT'18: 10th International Conference on Cryptology in Africa, Vol. 10831 of LNCS, Springer, 2018.
- [32] B. Smyth, A. Pironti, Truncating TLS Connections to Violate Beliefs in Web Applications, in: WOOT'13, USENIX Association, 2013.
- [33] V. Cortier, B. Smyth, Attacking and fixing Helios: An analysis of ballot secrecy, in: CSF'11, IEEE, 2011, pp. 297–311.
- [34] B. Smyth, V. Cortier, A note on replay attacks that violate privacy in electronic voting schemes, Tech. Rep. RR-7643, INRIA (2011).
- [35] B. Smyth, Replay attacks that violate ballot secrecy in helios, Technical Report 2012/185, Cryptology ePrint Archive (2012).
- [36] V. Cortier, B. Smyth, Attacking and fixing Helios: An analysis of ballot secrecy, *Journal of Computer Security* 21 (1) (2013) 89–148.

- [37] B. Smyth, Ballot secrecy: Security definition, sufficient conditions, and analysis of Helios, Technical Report 2015/942, Cryptology ePrint Archive (2018).
- [38] D. Bernhard, O. Pereira, B. Warinschi, How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios, in: ASIACRYPT'12, Vol. 7658 of LNCS, Springer, 2012, pp. 626–643.