Attacking and fixing Helios: An analysis of ballot secrecy

Véronique Cortier[1] and Ben Smyth[2]

[1]CNRS, Loria, UMR 7503 F-54506 Vandœuvre-lés-Nancy, France
[2]INRIA Paris-Rocquencourt, France

August 27, 2012

Abstract

Helios 2.0 is an open-source web-based end-to-end verifiable electronic voting system, suitable for use in low-coercion environments. In this article, we analyse ballot secrecy in Helios and discover a vulnerability which allows an adversary to compromise the privacy of voters. The vulnerability exploits the absence of ballot independence in Helios and works by replaying a voter's ballot or a variant of it, the replayed ballot magnifies the voter's contribution to the election outcome and this magnification can be used to violated privacy. We demonstrate the practicality of the attack by violating a voter's privacy in a mock election using the software implementation of Helios. Moreover, the feasibility of an attack is considered in the context of French legislative elections and, based upon our findings, we believe it constitutes a real threat to ballot secrecy. We present a fix and show that our solution satisfies a formal definition of ballot secrecy using the applied pi calculus. Furthermore, we present similar vulnerabilities in other electronic voting protocols – namely, the schemes by Lee *et al.*, Sako & Kilian, and Schoenmakers – which do not assure ballot independence. Finally, we argue that independence and privacy properties are unrelated, and non-malleability is stronger than independence.

Keywords. Applied Pi Calculus, Attack, Ballot Independence, Ballot Secrecy, Electronic Voting, Helios, Malleability, Privacy.

[†]This document is a preprint of a Journal of Computer Security article and a preliminary version was presented at the 24th IEEE Computer Security Foundations Symposium [CS11]. The research leading to these results were performed as part of the ProSecure project which is funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n° 258865, and the ANR-07-SeSur-002 AVOTÉ project. Smyth's work was predominantly done at CNRS, Loria, UMR 7503 F-54506 Vandœuvre-lés-Nancy, France and partly at Toshiba Research & Development Center, Kawasaki, Japan.

1 Introduction

Paper-based elections derive security properties from physical characteristics of the real-world. For example, marking a ballot in the isolation of a polling booth and depositing the completed ballot into a locked ballot box provides privacy; the polling booth also ensures that voters cannot be influenced by other votes and the locked ballot box prevents the announcement of early results, thereby ensuring fairness; and the transparency of the whole election process from ballot casting to tallying and the impossibility of altering the markings on a paper ballot sealed inside a locked ballot box gives an assurance of correctness and facilitates verifiability. Replicating these attributes in a digital setting has proven to be difficult and, hence, the provision of secure electronic voting systems is an active research topic.

Informally, *privacy* for electronic voting systems is characterised by the following requirements [KR05, DKR06, BHM08]:

- *Ballot secrecy.* A voter's vote is not revealed to anyone.
- *Receipt freeness.* A voter cannot gain information which can be used to prove, to a coercer, how she voted.
- *Coercion resistance.* A voter cannot collaborate, with a coercer, to gain information which can be used to prove how she voted.

Verifiability includes three properties [JCJ05, Dag07, KRS10]:

- *Individual verifiability.* A voter can check that her own ballot is published on the election's bulletin board.
- *Universal verifiability.* Anyone can check that all the votes in the election outcome correspond to ballots published on the election's bulletin board.
- *Eligibility verifiability.* Anyone can check that each ballot published on the bulletin board was cast by a registered voter and at most one ballot is tallied per voter.

Finally, *fairness* – summarised by the notion that *all voters are equal* – has not be thoroughly studied, but nonetheless we believe the following aspects are desirable:

- *Ballot independence.* Observing another voter's interaction with the election system does not allow a voter to cast a meaningfully related vote.
- *No early results.* A voter cannot change her vote once partial results are available.
- *Pulling out.* Once partial results are available a voter cannot abort.

The privacy property helps ensure that voters can express their free-will without fear of retribution, in particular, receipt freeness and coercion resistance

attempt to prevent intimidation of voters. In addition, receipt freeness helps prevent vote buying. The individual, universal and eligibility verifiability properties (also called *end-to-end verifiability* [JCJ02, CRS05, Adi06, Dag07, Adi08]) allow voters and election observers to verify – independently of the hardware and software running the election – that votes have been recorded, tallied and declared correctly. The fairness property prohibits the voting system from influencing a voter's behaviour, that is, observation of the voting system does not leak information that may affect a voter's vote, for example, ballot independence prevents Bob from casting the same vote as Alice (possibly without learning Alice's vote)[1]. In this article, we analyse ballot secrecy in Helios 2.0 [AMPQ09].

Formal definitions of ballot secrecy have been introduced in the context of the applied pi calculus by Delaune, Kremer & Ryan [KR05, DKR06, DKR09, DKR10] and Backes, Hriţcu & Maffei [BHM08]. These privacy definitions consider two voters $\mathcal{A}$, $\mathcal{B}$ and two candidates t, t'. Ballot secrecy is captured by the assertion that an adversary (controlling arbitrary many dishonest voters) cannot distinguish between a situation in which voter $\mathcal{A}$ votes for candidate t and voter $\mathcal{B}$ votes for candidate t', from another situation in which $\mathcal{A}$ votes t' and $\mathcal{B}$ votes t. This can be expressed by the following equivalence.

$$\mathcal{A}(t) \mid \mathcal{B}(t') \approx_l \mathcal{A}(t') \mid \mathcal{B}(t)$$

These formal definitions of ballot secrecy have been used by their respective authors to analyse the electronic voting protocols due to: Fujioka, Okamoto & Ohta [FOO92], Okamoto [Oka98], Lee *et al.* [LBD+04], and Juels, Catalano & Jakobsson [JCJ02, JCJ05, JCJ10]. It therefore seems natural to check whether Helios satisfies these formal definitions.

Helios 2.0. Helios is an open-source web-based electronic voting system which uses homomorphic encryption. The scheme is claimed to satisfy ballot secrecy [AMPQ09], but the nature of remote voting makes the possibility of satisfying stronger privacy properties difficult and Helios does not satisfy receipt freeness nor coercion resistance (satisfying these stronger privacy properties typically increases the voting system's complexity and, hence, a scheme satisfying ballot secrecy, rather than coercion resistance, may be preferred due its relative simplicity). In addition to ballot secrecy, the system provides individual and universal verifiability (cf. [KRS10, SRKK10] and [Smy11, Chapter 3] for an analysis of verifiability in Helios). Helios is particularly significant due to its real-world deployment: the International Association of Cryptologic Research (IACR) used Helios to elect its board members [BVQ10], following a successful trial in a non-binding poll [HBH10]; the Catholic University of Louvain adopted the system to elect the university president [AMPQ09]; and Princeton University used Helios to elect the student vice president [Pri10].

[1] Bulens, Giry & Pereira [BGP11, §3.2] question whether ballot independence is a desirable property of electronic voting systems and highlight the investigation of voting schemes which allow the submission of meaningfully related votes whilst preserving privacy as an interesting research direction.

1.1 Contribution

Our analysis of Helios reveals an attack which violates ballot secrecy. The attack exploits the system's lack of ballot independence, and works by replaying a voter's ballot or a variant of it (without knowing the vote contained within that ballot). Replaying a voter's ballot immediately violates ballot secrecy in an election with three voters. For example, consider an attack in an election with three voters – namely, Alice, Bob, and Mallory – as follows: if Mallory replays Alice's ballot, then Mallory can reveal Alice's vote by observing the election outcome and checking which candidate obtained at least two votes. The practicality of our attack has been demonstrated by violating a voter's privacy in a mock election using the software implementation of Helios. Furthermore, the vulnerability can be exploited in more realistic settings and, as an illustrative example, we discuss the feasibility of the attack in French legislative elections. This case study suggests there is a plausible threat to voters' privacy in elections using Helios. We also propose variants of the attack which abuse the malleability of ballots to ensure ballots cast by the adversary are distinct; this makes detecting the attack non-trivial (that is, checking for exact duplicates is insufficient to ensure ballot secrecy). Nonetheless, we fix the Helios protocol by identifying and discarding adversarial ballots. We believe this solution is particular well-suited because it maintains Benaloh's principle of ballot casting assurance [Ben06, Ben07] and requires a minimal extension to the Helios codebase. The revised scheme is shown to satisfy a formal definition of ballot secrecy using the applied pi calculus. In addition, we demonstrate that the absence of ballot independence can be exploited in other electronic voting protocols to violate privacy; in particular, a similar attack is shown against the protocol by Lee *et al.* [LBD+04] whereby an adversary replays a voter's ballot or a variant of it, and verbatim replay attacks are demonstrated against two schemes presented at CRYPTO (namely, the protocols due to Sako & Kilian [SK94] and Schoenmakers [Sch99]). Finally, we present some evidence to demonstrate that independence and privacy are unrelated properties, and non-malleability is stronger than independence.

Structure of this article. Section 2 presents the Helios electronic voting scheme. (We remark that this is the first cryptographic description of the Helios protocol in the literature and, hence, is an additional contribution of this article.) Section 3 describes our attack and some variants, in addition to a study of the attack's feasibility in the context of French legislative elections. We propose several solutions for recovering privacy in Section 4 and prove that our adopted solution formally satisfies ballot secrecy in Section 5. Section 6 demonstrates that the absence of ballot independence can be similarly exploited in other electronic voting protocols to violate privacy and Section 7 considers relationships between ballot independence and other security properties. Finally, Section 8 considers related work and our conclusion appears in Section 9.

2 Background: Helios 2.0

We provide a full description of Helios 2.0. This scheme exploits the additive homomorphic [CDS94, CGS97, Sch09] and distributed decryption [Ped91, CP93] properties of ElGamal [ElG85]. In addition, signature proofs of knowledge are used to ensure secrecy and integrity of the ElGamal scheme, and to ensure voters encrypt valid votes. We will recall these cryptographic primitives before presenting the Helios protocol.

2.1 Additive homomorphic ElGamal

Given cryptographic parameters (p, q, g) and a number $n \in \mathbb{N}$ of trustees, where p and q are large primes such that $q \mid p-1$ and g is a generator of the multiplicative group $\mathbb{Z}_p^*$ of order q, the following operations are defined by ElGamal.

Distributed key generation. Each trustee $i \in n$ selects a private key share $x_i \in_R \mathbb{Z}_q^*$ and computes a public key share $h_i = g^{x_i} \bmod p$. The public key is $h = h_1 \cdot \ldots \cdot h_n \bmod p$.

Encryption. Given a message m and a public key h, select a random nonce $r \in_R \mathbb{Z}_q^*$ and derive the ciphertext $(a, b) = (g^r \bmod p,\ g^m \cdot h^r \bmod p)$.

Re-encryption. Given a ciphertext (a, b) and public key h, select a random nonce $r' \in_R \mathbb{Z}_q^*$ and derive the re-encrypted ciphertext $(a', b') = (a \cdot g^{r'} \bmod p,\ b \cdot h^{r'} \bmod p)$.

Homomorphic addition. Given two ciphertexts (a, b) and (a', b'), the homomorphic addition of plaintexts is computed by multiplication $(a \cdot a' \bmod p,\ b \cdot b' \bmod p)$.

Distributed decryption. Given a ciphertext (a, b), each trustee $i \in n$ computes the partial decryption $k_i = a^{x_i}$. The plaintext $m = \log_g M$ is recovered from $M = b/(k_1 \cdot \ldots \cdot k_n) \bmod p$.

The computation of a discrete logarithm $\log_g M$ is hard in general. However, if M is chosen from a restricted domain, then the complexity is reduced, for example, if M is an integer such that $0 \leq M \leq n$, then the complexity is $O(n)$ by linear search or $O(\sqrt{n})$ using the baby-step giant-step algorithm [Sha71] (see also [LL90, §3.1]).

For secrecy, each trustee $i \in n$ must demonstrate knowledge of a discrete logarithm $\log_g h_i$, that is, they prove that h_i has been correctly constructed; this prevents, for example, a trustee constructing their public key share $h_i = h$. For integrity of decryption, each trustee $i \in n$ must demonstrate equality between discrete logarithms $\log_g h_i$ and $\log_a k_i$; this prevents, for example, a trustee constructing the public key share $h_i = g^{m+x_i}$ and providing the partial

decryption $k_i = a^{x_i}$. These proofs can be achieved using signatures of knowledge (see Appendix A for details). In addition, the voter must demonstrate that a valid vote has been encrypted and we describe a suitable signature of knowledge scheme in the following section.

2.2 Disjunctive proof of equality between discrete logs

Given the aforementioned cryptographic parameters (p, q, g), a signature of knowledge demonstrating that a ciphertext (a, b) contains either 0 or 1 (without revealing which), can be constructed by proving that either $\log_g a = \log_h b$ or $\log_g a = \log_h b/g^m$, that is, by application of a signature of knowledge demonstrating a disjunctive proof of equality between discrete logarithms [CDS94, Sch09]. Observe for a valid ciphertext (a, b) that $a \equiv g^r \bmod p$ and $b \equiv h^r \cdot g^m \bmod p$ for some nonce $r \in \mathbb{Z}_q^*$, hence the former disjunct $\log_g g^r = \log_h h^r \cdot g^m$ is satisfied when $m = 0$, and the latter disjunct $\log_g g^r = \log_h (h^r \cdot g^m)/g^m$ is satisfied when $m = 1$. This technique is generalised by Adida *et al.* [AMPQ09] to allow a signature of knowledge demonstrating that a ciphertext (a, b) contains message m, where $m \in \{\mathsf{min}, \ldots, \mathsf{max}\}$ for some system parameters $\mathsf{min} \in \mathbb{N}$ and $\mathsf{max} \in \mathbb{N}^*$ such that $\mathsf{min} \leq \mathsf{max}$. Formally, a signature of knowledge demonstrating a disjunctive proof of equality between discrete logarithms can be derived, and verified, as follows [AMPQ09, CDS94, Sch09], where $\mathcal{H}$ is a SHA-256 hash function.

Sign. Given ciphertext (a, b) such that $a \equiv g^r \bmod p$ and $b \equiv h^r \cdot g^m \bmod p$ for some nonce $r \in \mathbb{Z}_q^*$, where plaintext $m \in \{\mathsf{min}, \ldots, \mathsf{max}\}$. For all $i \in \{\mathsf{min}, \ldots, m-1, m+1, \ldots, \mathsf{max}\}$, compute challenge $c_i \in_R \mathbb{Z}_q^*$, response $s_i \in_R \mathbb{Z}_q^*$ and witnesses $a_i = g^{s_i}/a^{c_i} \bmod p$ and $b_i = h^{s_i}/(b/g^i)^{c_i} \bmod p$. Select a random nonce $w \in_R \mathbb{Z}_q^*$. Compute witnesses $a_m = g^w \bmod p$ and $b_m = h^w \bmod p$, challenge $c_m = \mathcal{H}(a_{\mathsf{min}}, b_{\mathsf{min}}, \ldots, a_{\mathsf{max}}, b_{\mathsf{max}}) - \sum_{i \in \{\mathsf{min}, \ldots, m-1, m+1, \ldots, \mathsf{max}\}} c_i \pmod q$ and response $s_m = w + r \cdot c_m \bmod q$.

Verify. Given (a, b) and $(a_{\mathsf{min}}, b_{\mathsf{min}}, c_{\mathsf{min}}, s_{\mathsf{min}}, \ldots, a_{\mathsf{max}}, b_{\mathsf{max}}, c_{\mathsf{max}}, s_{\mathsf{max}})$, for each $\mathsf{min} \leq i \leq \mathsf{max}$ check $g^{s_i} \equiv a_i \cdot a^{c_i} \pmod p$ and $h^{s_i} \equiv b_i \cdot (b/g^i)^{c_i} \pmod p$. Finally, check $\mathcal{H}(a_{\mathsf{min}}, b_{\mathsf{min}}, \ldots, a_{\mathsf{max}}, b_{\mathsf{max}}) \equiv \sum_{\mathsf{min} \leq i \leq \mathsf{max}} c_i \pmod q$.

A valid proof asserts that (a, b) is a ciphertext containing the message m such that $m \in \{\mathsf{min}, \ldots, \mathsf{max}\}$.

2.3 Protocol description

An election is created by naming an election officer, selecting a set of trustees, and generating a distributed public key pair. The election officer publishes, on the bulletin board, the public part of the trustees' key (and proof of correct construction), the candidate list $\tilde{t} = (t_1, \ldots, t_\ell) \cup \{\epsilon\}$ (where ϵ represents a vote of abstention), and the list of eligible voters $\widetilde{id} = (id_1, \ldots, id_n)$; the officer

also publishes the *election fingerprint*, that is, the hash of these parameters. Informally, the steps that participants take during a run of Helios are as follows.

1. The voter launches a browser script that downloads the election parameters and recomputes the election fingerprint. The voter should verify that the fingerprint corresponds to the value published on the bulletin board. (This ensures that the script is using the trustees' public key, in particular, it helps prevent encrypting a vote with an adversary's public key. Such attacks have been discussed in the context of Direct Anonymous Attestation by Rudolph [Rud07]; although, the vulnerability was discounted, in the trusted computing setting, by Leung, Chen & Mitchell [LCM08].)

2. The voter inputs her vote $v \in \tilde{t}$ to the browser script, which creates a ballot consisting of her vote encrypted by the trustees' public key, and a proof that the ballot represents a permitted vote (this is needed because the ballots are never decrypted individually, in particular, it prevents multiple votes being encoded as a single ballot). The ballot is displayed to the voter.

3. The voter can audit the ballot to check if it really represents a vote for her chosen candidate; if she decides to do this, then the script provides her with the random data used in the ballot creation. She can then independently reconstruct her ballot and verify that it is indeed well-formed. The script provides some practical resistance against vote selling by refusing to cast audited ballots. See Benaloh [Ben06, Ben07] for further details on ballot auditing.

4. When the voter has decided to cast her ballot, the script submits it to the election officer. The election officer authenticates the voter and checks that she is eligible to vote. The election officer also verifies the proof and publishes the ballot, appended with the voter's identity id, on the bulletin board. (In practice, the election officer also publishes the hash of the ballot, we omit this detail for brevity.)

5. Individual voters can check that their ballots appear on the bulletin board and, by verifying the proof, observers are assured that ballots represent permitted votes.

6. After some predefined deadline, the election officer homomorphically combines the ballots and publishes the encrypted tally on the bulletin board. Anyone can check that tallying is performed correctly.

7. Each of the trustees publishes a partial decryption of the encrypted tally, together with a signature of knowledge proving the partial decryption's correct construction. Anyone can verify these proofs.

8. The election officer decrypts the tally and publishes the result. Anyone can check this decryption.

Figure 1 Ballot construction by the browser script

Input: Cryptographic parameters (p, q, g), public key h, candidate list $\tilde{t} = (t_1, \ldots, t_\ell) \cup \{\epsilon\}$ and vote v.

Output: Encrypted vote $(a_1, b_1), \ldots, (a_\ell, b_\ell)$, signatures of knowledge $(\bar{a}_1, \bar{b}_1, \bar{c}_1, \bar{s}_1, \bar{a}'_1, \bar{b}'_1, \bar{c}'_1, \bar{s}'_1), \ldots, (\bar{a}_\ell, \bar{b}_\ell, \bar{c}_\ell, \bar{s}_\ell, \bar{a}'_\ell, \bar{b}'_\ell, \bar{c}'_\ell, \bar{s}'_\ell)$ and signature of knowledge $(\bar{a}, \bar{b}, \bar{c}, \bar{s}, \bar{a}', \bar{b}', \bar{c}', \bar{s}')$, where the signatures are constructed using the algorithm presented in Section 2.2.

1. If $v \notin \tilde{t}$ then the script terminates.
2. Encode the vote v as a bitstring. For all $1 \leq i \leq \ell$, let
$$m_i = \begin{cases} 1 & \text{if } v = t_i \\ 0 & \text{otherwise} \end{cases}$$
3. The bitstring representing the vote is encrypted. For all $1 \leq i \leq \ell$, let
$$(a_i, b_i) = (g^{r_i} \bmod p,\ g^{m_i} \cdot h^{r_i} \bmod p)$$
where $r_i \in_R \mathbb{Z}_q^*$.
4. For all $1 \leq i \leq \ell$, let $(\bar{a}_i, \bar{b}_i, \bar{c}_i, \bar{s}_i, \bar{a}'_i, \bar{b}'_i, \bar{c}'_i, \bar{s}'_i)$ be a signature of knowledge demonstrating that the ciphertext (a_i, b_i) contains either 0 or 1, that is, each candidate can receive at most one vote.
5. Let $(\bar{a}, \bar{b}, \bar{c}, \bar{s}, \bar{a}', \bar{b}', \bar{c}', \bar{s}')$ be a signature of knowledge demonstrating that the ciphertext $(a_1 \cdot \ldots \cdot a_\ell,\ b_1 \cdot \ldots \cdot b_\ell)$ contains either 0 or 1, that is, at most one candidate receives one vote.

Formally, Step 2 is defined in Figure 1. (For simplicity the ballot construction algorithm in Figure 1 considers a vote $v \in \tilde{t}$, this can be generalised [AMPQ09] to consider a vote $\tilde{v} \subseteq \tilde{t}$.) Checking voter eligibility (Step 4) is beyond the scope of Helios and Adida *et al.* [AMPQ09] propose the use of existing infrastructure. The remaining steps follow immediately from the application of cryptographic primitives (see Section 2.1 for details).

2.4 Software implementation

Helios 3.0 is an extension of Helios 2.0 which adds numerous practical features, including: integration of authentication with various web-services (for example, Facebook, GMail and Twitter), bulk voter registration using pre-existing electoral rolls, and simplification of administration with multiple trustees. Helios 3.0 has been implemented and is publicly available: `http://heliosvoting.org/`.

3 Attacking ballot secrecy

Ballot secrecy means "a voter's vote is not revealed to anyone" and this section shows that Helios does not satisfy this definition by presenting an attack which allows an adversary to reveal a voter's vote (Section 5 will show that formal definitions of ballot secrecy [KR05, DKR09, BHM08] are also violated). Intuitively, an adversary may identify a voter's ballot on the bulletin board (using the voter's identity id) and recast this ballot by corrupting dishonest voters. The multiple occurrences of the voter's ballot will magnify the voter's contribution to the election outcome, thereby leaking information that can be exploited to violate the voter's privacy. The remainder of this section proceeds as follows: a description of the attack for three voters appears in Section 3.1 and variants are considered in Section 3.2, the attack is generalised to arbitrary many voters in Section 3.3 and the threat to real elections is also considered.

3.1 Attack description

Let us consider an election with candidates $t_1, \ldots, t_\ell$ and three eligible voters who have identities id_1, id_2 and id_3. Suppose that voters id_1 and id_2 are honest, and id_3 is a dishonest voter controlled by the adversary. Further assume that the honest voters have cast their ballots. The bulletin board entries are as follows:

$$\begin{array}{l} id_1, ciph_1, spk_1, spk'_1 \\ id_2, ciph_2, spk_2, spk'_2 \end{array}$$

where for $i \in \{1, 2\}$ we have

$$\begin{array}{lcl} ciph_i & = & (a_{i,1}, b_{i,1}), \ldots, (a_{i,\ell}, b_{i,\ell}) \\ spk_i & = & (\bar{a}_{i,1}, \bar{b}_{i,1}, \bar{c}_{i,1}, \bar{s}_{i,1}, \bar{a}'_{i,1}, \bar{b}'_{i,1}, \bar{c}'_{i,1}, \bar{s}'_{i,1}), \\ & & \quad \ldots, (\bar{a}_{i,\ell}, \bar{b}_{i,\ell}, \bar{c}_{i,\ell}, \bar{s}_{i,\ell}, \bar{a}'_{i,\ell}, \bar{b}'_{i,\ell}, \bar{c}'_{i,\ell}, \bar{s}'_{i,\ell}) \\ spk'_i & = & (\bar{a}_i, \bar{b}_i, \bar{c}_i, \bar{s}_i, \bar{a}'_i, \bar{b}'_i, \bar{c}'_i, \bar{s}'_i) \end{array}$$

The value $ciph_i$ is the ith voter's encrypted vote, spk_i demonstrates that ciphertexts $(a_{i,1}, b_{i,1}), \ldots, (a_{i,\ell}, b_{i,\ell})$ contain either 0 or 1 (that is, the voter has assigned at most one vote to each candidate), and spk'_i demonstrates that $(a_{i,1} \cdot \ldots \cdot a_{i,\ell},\ \ b_{i,1} \cdot \ldots \cdot b_{i,\ell})$ contains either 0 or 1 (that is, the voter has voted for at most one candidate).

Replaying a ballot. The adversary observes the bulletin board and selects $ciph_k, spk_k, spk'_k$ such that id_k is the voter whose privacy will be compromised, where $k \in \{1, 2\}$. The adversary casts the ballot $ciph_k, spk_k, spk'_k$ and it immediately follows that the bulletin board is composed as follows:

$$\begin{array}{l} id_1, ciph_1, spk_1, spk'_1 \\ id_2, ciph_2, spk_2, spk'_2 \\ id_3, ciph_k, spk_k, spk'_k \end{array}$$

It is trivial to observe that each bulletin board entry represents a permitted vote, that is, $spk_1, spk'_1, spk_2, spk'_2, spk_k, spk'_k$ all contain valid signatures of knowledge. It follows that Helios does not satisfy ballot independence: observing another voter's interaction with the election system allows a voter to cast the *same* vote. The absence of ballot independence will now be exploited to violate privacy.

Violating privacy. The homomorphic addition of ballots reveals the encrypted tally $(a_{1,1} \cdot a_{2,1} \cdot a_{k,1},\ b_{1,1} \cdot b_{2,1} \cdot b_{k,1}), \ldots, (a_{1,\ell} \cdot a_{2,\ell} \cdot a_{k,\ell},\ b_{1,\ell} \cdot b_{2,\ell} \cdot b_{k,\ell})$ and, given the partial decryptions, these ciphertexts can be decrypted to reveal the number of votes for each candidate. Since there will be at least two votes for the candidate voter id_k voted for, the voter's vote can be revealed and hence privacy is not preserved. Moreover, the vote of the remaining honest voter will also be revealed.

A video demonstrating the attack against the Helios 3.0 implementation has been produced [SC10].

In the aforementioned attack description, the ballots cast by two voters are identical. This behaviour is not detected by Helios and, prior to our work, human detection – for example, by auditing – would have been improbable. Of course, further to our results, the aforementioned attack can be detected by searching for duplicated ballots. For a covert attack, the adversary may replay ballots in different elections, when the trustees' public key is reused and the candidate lists for each election are of equal length. However, this is not generally possible in Helios since fresh keys should be used for each election. The following section introduces further variants of our attack that exploit malleability to derive distinct ballots, thereby demonstrating that searching for duplicate ballots is insufficient to ensure ballot secrecy.

3.2 Variants exploiting ballot malleability

Let us consider variants of our attack under the assumptions presented in the previous section: we have an election with candidates $t_1, \ldots, t_\ell$ and three eligible voters such that the two honest voters have cast their votes using identities id_1 and id_2, and the remaining dishonest voter is controlled by the adversary, where the dishonest voter has the identity id_3. Given that we will consider ballot malleability, we refine our notation and consider the bulletin board entries of honest voters as follows:

$$id_1, ciph_{1,1}, \ldots, ciph_{1,\ell}, spk_{1,1}, \ldots, spk_{1,\ell}, spk'_1$$
$$id_2, ciph_{2,1}, \ldots, ciph_{2,\ell}, spk_{2,1}, \ldots, spk_{2,\ell}, spk'_2$$

where for all $i \in \{1, 2\}$ and $j \in \{1, \ldots, \ell\}$ we have

$$\begin{array}{lcl} ciph_{i,j} & = & (a_{i,j}, b_{i,j}) \\ spk_{i,j} & = & (\bar{a}_{i,j}, \bar{b}_{i,j}, \bar{c}_{i,j}, \bar{s}_{i,j}, \bar{a}'_{i,j}, \bar{b}'_{i,j}, \bar{c}'_{i,j}, \bar{s}'_{i,j}) \\ spk'_i & = & (\bar{a}_i, \bar{b}_i, \bar{c}_i, \bar{s}_i, \bar{a}'_i, \bar{b}'_i, \bar{c}'_i, \bar{s}'_i) \end{array}$$

The value $ciph_{i,j}$ is the ith voter's encrypted vote for the candidate t_j (that is, $ciph_{i,j}$ is a ciphertext containing the plaintext 1 if the voter voted t_j, and 0 otherwise), $spk_{i,j}$ demonstrates that the ciphertext $ciph_{i,j}$ contains either 0 or 1, and spk'_i is defined as before, namely, it demonstrates that $(a_{i,1} \cdot \ldots \cdot a_{i,\ell},\ b_{i,1} \cdot \ldots \cdot b_{i,\ell})$ contains either 0 or 1. In the remainder of this section we shall assume that the voter under attack casts the Ballot B0, namely,

$$id_k, ciph_{k,1}, \ldots, ciph_{k,\ell}, spk_{k,1}, \ldots, spk_{k,\ell}, spk'_k \tag{B0}$$

where $k \in \{1, 2\}$.

3.2.1 Integer representation attack

Given the Ballot B0, the adversary selects integers $r_1, r'_1, \ldots, r_\ell, r'_\ell, r, r' \in \mathbb{N}$ and constructs the following related ballot, namely,

$$ciph_{k,1}, \ldots, ciph_{k,\ell}, \overline{spk}_{k,1}, \ldots, \overline{spk}_{k,\ell}, \overline{spk}'_k \tag{B1}$$

where $\overline{spk}'_k = (\bar{a}_k, \bar{b}_k, \bar{c}_k, \bar{s}_k + r \cdot q, \bar{a}'_k, \bar{b}'_k, \bar{c}'_k, \bar{s}'_k + r' \cdot q)$ and for all $j \in \{1, \ldots, \ell\}$ we have $\overline{spk}_{k,j} = (\bar{a}_{k,j}, \bar{b}_{k,j}, \bar{c}_{k,j}, \bar{s}_{k,j} + r_j \cdot q, \bar{a}'_{k,j}, \bar{b}'_{k,j}, \bar{c}'_{k,j}, \bar{s}'_{k,j} + r'_j \cdot q)$. Ballot B1 adds multiples of q to the response components of Ballot B0, this changes the ballot but not the vote, because the ciphertexts that encrypt the vote remain unchanged. It follows that Ballot B1 can be cast by the adversary as a vote for the same candidate as the voter with identity id_k selected and privacy can be violated as described in Section 3.1. This might be considered an oversight, rather than a theoretical issue, because the ballots are identical if considered as group elements.

3.2.2 Permutation attacks

Given the Ballot B0, the adversary selects a permutation π on $\{1, \ldots, \ell\}$, where π is not the identity, and proceeds as follows.

Constructing a related ballot. The adversary constructs the Ballot B2:

$$ciph_{k,\pi(1)}, \ldots, ciph_{k,\pi(\ell)}, spk_{k,\pi(1)}, \ldots, spk_{k,\pi(\ell)}, spk'_k \tag{B2}$$

Ballot B2 permutes the ciphertexts included in Ballot B0, thereby deriving a ballot for a different candidate (with the exception of an abstention vote). The adversary can cast Ballot B2 and it is trivial to witness that this ballot will be accepted by the bulletin board, since $spk_{k,\pi(1)}, \ldots, spk_{k,\pi(\ell)}, spk'_k$ are all valid signatures of knowledge. It follows that we have shown another technique to violate ballot independence in Helios: observing another voter's interaction with the election system allows a voter to cast a *different* vote, with the exception of abstention votes which will be the *same*. (The ability to cast different votes may be of independent interest, for example, a voter can cast a distinct vote from their boss.) The absence of ballot independence can be exploited to violate privacy.

Violating privacy. The decrypted tally reveals the number of votes for each candidate and this data can be used to discover how each voter voted. First, if the tally contains votes for three distinct candidates, then there exists integers $i, j \in \{1, \ldots, \ell\}$ such that $\pi(i) = j$ and the voter with identity id_k voted for candidate t_i. Secondly, if the tally contains two votes for a candidate and one vote for another candidate, then the voter with identity id_k voted for the candidate with two votes. Finally, in the case where the outcome is unanimous, every vote is revealed[2]

3.2.3 Malformed ciphertext attack

Given the Ballot B0, the adversary can select an integer $v \in \{1, \ldots, \ell\}$ and proceed as follows.

Constructing a related ballot. The adversary constructs the Ballot B3, namely,

$$\underbrace{(1,1),\ldots,(1,1)}_{v-1 \text{ times}},(a_{k,v},b_{k,v}),\underbrace{(1,1),\ldots,(1,1)}_{\ell-v \text{ times}},$$
$$\overline{spk}_1,\ldots,\overline{spk}_{v-1},spk_{k,v},\overline{spk}_{v+1},\ldots,\overline{spk}_{\ell},spk_{k,v} \quad \text{(B3)}$$

such that for all $j \in \{1, \ldots, v-1, v+1, \ldots, \ell\}$ we have $\overline{spk}_j = (\hat{a}_j, \hat{b}_j, \hat{c}_j, \hat{s}_j, \hat{a}'_j, \hat{b}'_j, \hat{c}'_j, \hat{s}'_j)$ where $\hat{c}'_j, \hat{s}'_j, \hat{s}_j \in_R \mathbb{Z}_q^*$ and

$$\begin{aligned}
\hat{a}_j &= g^{\hat{s}_j} \bmod p \\
\hat{a}'_j &= g^{\hat{s}'_j} \bmod p \\
\hat{b}_j &= h^{\hat{s}_j} \bmod p \\
\hat{b}'_j &= h^{\hat{s}'_j} \cdot g^{\hat{c}'_j} \bmod p \\
\hat{c}_j &= \mathcal{H}(\hat{a}_j, \hat{b}_j, \hat{a}'_j, \hat{b}'_j) - \hat{c}'_j \bmod q
\end{aligned}$$

By definition of Ballot B0, it is trivial to witness that $spk_{k,v}$ is a valid proof for $(a_{k,v}, b_{k,v})$ and, therefore, $spk_{k,v}$ is a valid proof for the homomorphic combination of ciphertexts encapsulated in Ballot B3. Moreover, the following lemma demonstrates that for all $j \in \{1, \ldots, v-1, v+1, \ldots, \ell\}$ we have $\overline{spk}_j$ is valid proof for $(1, 1)$. We stress that Lemma 1 does not violate the soundness property of our signature of knowledge scheme for disjunctive proofs of equality between discrete logs (Section 2.2) because $\log_g a = \log_h b$ or $\log_g a = \log_h b/g^m$ holds for $(a, b) = (g^0, h^0)$.

Lemma 1. *The signature* $(\bar{a}, \bar{b}, \bar{c}, \bar{s}, \bar{a}', \bar{b}', \bar{c}', \bar{s}')$ *is valid for* $(1, 1)$*, where* $\bar{c}', \bar{s}', \bar{s} \in_R \mathbb{Z}_q^*$*,* $\bar{a} = g^{\bar{s}} \pmod p$*,* $\bar{a}' = g^{\bar{s}'} \pmod p$*,* $\bar{b} = h^{\bar{s}} \pmod p$*,* $\bar{b}' = h^{\bar{s}'} \cdot g^{\bar{c}'} \pmod p$*, and* $\bar{c} = \mathcal{H}(\bar{a}, \bar{b}, \bar{a}', \bar{b}') - \bar{c}' \pmod q$*.*

[2]Unanimous election results highlight an inadequacy in our informal definition of privacy: as stated (Section 1), our definition is unsatisfiable. This issue is overcome in our formal privacy definition (Section 5.3).

Proof. Suppose the signature $(\bar{a}, \bar{b}, \bar{c}, \bar{s}, \bar{a}', \bar{b}', \bar{c}', \bar{s}')$ is defined above and let $(a, b) = (1, 1)$. We must show that (a, b) and $(\bar{a}, \bar{b}, \bar{c}, \bar{s}, \bar{a}', \bar{b}', \bar{c}', \bar{s}')$ satisfy the conditions of the verification algorithm described in Section 2.2. Since $a^{\bar{c}} = 1$ and $(b/g^0)^{\bar{c}} = 1$, we trivially derive $g^{\bar{s}} \equiv \bar{a} \cdot a^{\bar{c}} \pmod{p}$ and $h^{\bar{s}} \equiv \bar{b} \cdot (b/g^0)^{\bar{c}} \pmod{p}$. Moreover, since $a^{\bar{c}'} = 1$ and $(b/g^1)^{\bar{c}'} = g^{-\bar{c}'}$, it follows that $g^{\bar{s}'} \equiv \bar{a}' \cdot a^{\bar{c}'} \pmod{p}$ and $h^{\bar{s}'} \equiv \bar{b}' \cdot (b/g^1)^{\bar{c}'} \pmod{p}$. Finally, recall $\bar{c} = \mathcal{H}(\bar{a}, \bar{b}, \bar{a}', \bar{b}') - \bar{c}' \pmod{q}$ and therefore $\mathcal{H}(\bar{a}, \bar{b}, \bar{a}', \bar{b}') \equiv \bar{c} + \bar{c}' \pmod{q}$, concluding our proof. □

It follows immediately that the adversary's Ballot B3 will be accepted by the bulletin board, hence, we have shown another technique to violate ballot independence in Helios: observing another voter's interaction with the election system allows a voter to cast a *meaningfully related* vote, in particular, if a voter votes for candidate t_v or ϵ, then the *same* vote can be cast, otherwise, a *different* vote (namely, ϵ) can be cast. The absence of ballot independence can be exploited to violate privacy.

Violating privacy. The homomorphic addition of ballots reveals the encrypted tally $(A_1, B_1), \ldots, (A_\ell, B_\ell)$ defined as follows:

$$\begin{aligned}(a_{1,1} \cdot a_{2,1},\ b_{1,1} \cdot b_{2,1}), \ldots, (a_{1,v-1} \cdot a_{2,v-1},\ b_{1,v-1} \cdot b_{2,v-1}),\\ (a_{1,v} \cdot a_{2,v} \cdot a_{k,v},\ b_{1,v} \cdot b_{2,v} \cdot b_{k,v}),\\ (a_{1,v+1} \cdot a_{2,v+1},\ b_{1,v+1} \cdot b_{2,v+1}), \ldots, (a_{1,\ell} \cdot a_{2,\ell},\ b_{1,\ell} \cdot b_{2,\ell})\end{aligned}$$

Given the partial decryptions, the tally can be decrypted to reveal the number of votes for each candidate. If the tally contains two votes for some candidate, one vote for some other candidate, and no votes for abstention, then the honest voter with identity id_k must have cast a vote for the candidate with two votes and hence privacy is not preserved. A straightforward derivative of this attack allows privacy to be violated when candidate t_v receives one vote, since the adversary learns that the voter with identity id_k did not vote for this candidate and therefore must have voted for the candidate with the remaining vote.

3.2.4 Homomorphic attack

Following from the methodology introduced by the malformed ciphertext attack (Section 3.2.3), we propose an attack that allows an adversary to construct a ballot related to an abstention vote cast by an honest voter. The attack proceeds as follows.

Constructing a related ballot. The adversary constructs the Ballot B4, namely,

$$(a_{k,1} \cdot \ldots \cdot a_{k,\ell},\ b_{k,1} \cdot \ldots \cdot b_{k,\ell}), \underbrace{(1,1), \ldots, (1,1)}_{\ell - 1 \text{ times}}, spk'_k, \overline{spk}_2, \ldots, \overline{spk}_\ell, spk'_k \quad \text{(B4)}$$

where for all $2 \leq i \leq \ell$ the signature $\overline{spk}_i$ is constructed in accordance with the definition given in Lemma 1. It follows immediately for all $2 \leq i \leq \ell$ that

$\overline{spk}_i$ is a valid proof for $(1,1)$. Moreover, by definition of Ballot B0, it is trivial to witness that spk'_k is a valid proof for $(a_{k,1} \cdot \ldots \cdot a_{k,\ell},\ b_{k,1} \cdot \ldots \cdot b_{k,\ell})$ and, therefore, spk'_k is a valid proof for the homomorphic combination of ciphertexts encapsulated in the Ballot B4. It follows that the adversary's Ballot B4 will be accepted by the bulletin board, hence, we have shown another technique to cast a *meaningfully related* vote, in particular, if a voter cast an abstention vote, then the *same* vote can be cast, otherwise, a *different* vote (namely, a vote for candidate t_1) can be cast. The absence of ballot independence can again be exploited to violate privacy.

Violating privacy. The homomorphic addition of ballots reveals the encrypted tally $(a_{1,1} \cdot a_{2,1} \cdot a_{k,1} \cdot \ldots \cdot a_{k,\ell},\ b_{1,1} \cdot b_{2,1} \cdot b_{k,1} \cdot \ldots \cdot b_{k,\ell}), (a_{1,2} \cdot a_{2,2},\ b_{1,2} \cdot b_{2,2}) \ldots, (a_{1,\ell} \cdot a_{2,\ell},\ b_{1,\ell} \cdot b_{2,\ell})$. If the decrypted tally contains two votes for abstention and one vote for some candidate, then the voter with identity id_k cast a vote for abstention and hence privacy is not preserved.

3.2.5 Further malleability attacks

A further variant that exploits malleability has been introduced by Bernhard [Ber12] and, concurrently, Desmedt & Chaidos [DC12a, DC12b]. We recall the details here for completion. Given the Ballot B0, the adversary selects $r_1, \ldots, r_\ell \in \mathbb{N}$ and constructs the Ballot B5, namely,

$$\overline{ciph}_{k,1}, \ldots, \overline{ciph}_{k,\ell}, \overline{spk}_{k,1}, \ldots, \overline{spk}_{k,\ell}, \overline{spk}'_k \tag{B5}$$

such that for all $j \in \{1, \ldots, \ell\}$ we have

$$\begin{array}{lcl} \overline{ciph}_{k,j} & = & (a_{k,j} \cdot g^{r_j},\ b_{k,j} \cdot h^{r_j}) \\ \overline{spk}_{k,j} & = & (\bar{a}_{k,j}, \bar{b}_{k,j}, \bar{c}_{k,j}, \bar{s}_{k,j} + r_j \cdot c_{k,j}, \bar{a}'_{k,j}, \bar{b}'_{k,j}, \bar{c}'_{k,j}, \bar{s}'_{k,j} + r_j \cdot \bar{c}_{k,j}) \\ \overline{spk}'_k & = & (\bar{a}_k, \bar{b}_k, \bar{c}_k, \bar{s}_k + r \cdot \bar{c}_k, \bar{a}'_k, \bar{b}'_k, \bar{c}'_k, \bar{s}'_k + r \cdot \bar{c}'_k) \end{array}$$

where $r = r_1 + \ldots + r_\ell$. As shown by Bernhard [Ber12], the signatures of knowledge $\overline{spk}_{k,1}, \ldots, \overline{spk}_{k,\ell}, \overline{spk}'_k$ are valid (Lemma 2).

Lemma 2. *If $(\bar{a}, \bar{b}, \bar{c}, \bar{s}, \bar{a}', \bar{b}', \bar{c}', \bar{s}')$ is a signature for (a, b), then $(\bar{a}, \bar{b}, \bar{c}, \bar{s} + r \cdot \bar{c}, \bar{a}', \bar{b}', \bar{c}', \bar{s}' + r \cdot \bar{c}')$ is a signature for $(a \cdot g^r,\ b \cdot h^r)$, where $r \in \mathbb{N}$.*

It follows that the adversary's ballot (B5) will be accepted by the bulletin board and privacy can be violated as described in Section 3.1.

3.2.6 Attacks against the software implementation of Helios

The variants described in Sections 3.2.1 – 3.2.5 have been successfully launched against the Helios 3.0 implementation. Moreover, given Ballot B0 as input, a PHP script has been written to construct each of the related Ballots B1 – B5.

3.3 Generalised attack and French election case study

Our attack (Section 3.1) demonstrates that the ballot of an arbitrary voter can be replayed by any other voter and we have shown how privacy can be violated in elections with three voters. However, in general, the attack does not apply to elections with more than three voters, nonetheless, some information is leaked, and colluding voters can replay sufficiently many ballots to violate a voter's privacy. We will now discuss the feasibility of compromising ballot secrecy in a real-world election, focusing on the cost of an attack in French legislative elections, where each district elects a representative for the French National Assembly. (We limit discussion to the replay attack described in Section 3.1 for simplicity and stress that the variants of our attack presented in Section 3.2 can be similarly used in elections with more than three voters.) Districts have several polling stations and each polling station individually announces its tally [Fr]; these tallies are published in local newspapers. The publication of tallies is typical of French elections at all levels, for example, from the election of mayor, to the presidential election.

In this (standard) voting configuration, an adversary can violate the ballot secrecy of a given voter by corrupting voters registered at the same polling station (for example, a coalition of neighbours or a family). The corrupted voters replay the ballot of the voter under attack, as previously explained. The motivation for restricting the selection of corrupted voters to the same polling station is twofold. Firstly, fewer corrupt voters are required to significantly influence the tally of an individual polling station (in comparison to influencing the election outcome). Secondly, it is unlikely to change the district's elected representative, because a candidate will receive only a few additional votes in the district, it follows that coercing voters to sacrifice their vote, for the purposes of the attack, should be easier. In the remainder of this section, we discuss how many corrupt voters are required to violate ballot secrecy – by making a significant change in the tally of a polling station – in an arbitrary district of Aulnay-sous-Bois and a rural district in Toul.

3.3.1 Ballot secrecy in Aulnay-sous-Bois

Using historic data and/or polls, it is possible to construct the expected distribution of votes. For simplicity, let us assume the distribution of votes per polling station is the average of the 2010 tally (Table 1), and that if the adversary can increase the number of votes for a particular candidate by more than σ (by replaying a voter's ballot), then this is sufficient to determine that the voter voted for that candidate. In addition, suppose that the adversary corrupts abstaining voters and therefore we do not consider the redistribution of votes. We remark that corrupting abstaining voters may be a fruitful strategy, since abstaining voters do not sacrifice their vote by participating in an attack.

Table 2 presents the expected distribution of votes, and includes the number of voters that an adversary must corrupt to determine if a voter voted for a particular candidate, for various values of σ. We shall further assume that

Party	Tally
PS	4120
UMP	3463
FN	1933
Europe Eco.	1921
Front de gauche	880
NPA	697
MODEM	456
Debout la République	431
Alliance école	193
LO	156
Émergence	113
Liste chrétienne	113

Table 1: 2010 legislative election results in Aulnay-sous-Bois [Fr10]

participation in the region is consistent with 2010, that is, 291 of the 832 eligible voters are expected to participate. It follows that 50 voters corresponds to approximately 6% of the Aulnay-sous-Bois electorate, and 10 voters corresponds to approximately 1%. Our results therefore demonstrate that the privacy of a voter can be compromised by corrupting a small number of voters. In particular, for medium-size parties (in terms of votes received) – including, for example, FN and Europe Ecologie – it is sufficient to corrupt 19 voters to see the number of votes increase by 50%. Furthermore, given the low turn-out (541 voters are expected to abstain), it seems feasible to corrupt abstaining voters, and therefore an attack can be launched without any voter sacrificing their vote.

Limitations. For such an attack based upon a statistical model, we acknowledge that this model is rather naïve. For example, the attacker can never be certain that the distribution of votes follows from a previous election or a poll, in particular, differences may arise from changes in voter behaviour. Nevertheless, we believe our model is sufficiently indicative to illustrate the real threat of an attack against privacy. A definitive mathematical analysis could be considered in the future.

Cases of complete privacy breach. The probabilistic nature of these attacks may introduce sufficient uncertainty to prevent privacy violations, and we will consider voting configurations where an adversary can definitively learn a voter's vote. Observe that if an attacker can corrupt half of the voters at a polling station, then the vote of an arbitrary voter can be revealed. Moreover, the cost of this attack can be reduced. In particular, if n dishonest voter's replay voter $\mathcal{V}$'s ballot, then it is possible to deduce that $\mathcal{V}$ did not vote for any candidate that received strictly less than $n + 1$ votes. This leaks information about voter $\mathcal{V}$'s chosen candidate and in cases where exactly one candidate received

Party	Expected tally	$\sigma = 200\%$	$\sigma = 150\%$	$\sigma = 50\%$	$\sigma = 20\%$
PS	81	162	122	41	17
UMP	68	136	102	34	14
FN	38	76	57	19	8
Europe Eco.	38	76	57	19	8
Front de gauche	17	34	26	9	4
NPA	14	28	21	7	3
MODEM	9	18	14	5	2
Debout la République	8	16	12	4	2
Alliance école	4	8	6	2	1
LO	3	6	5	2	1
Émergence	2	4	3	1	1
Liste chrétienne	2	4	3	1	1

Table 2: Number of duplicate ballots for a significant change in the tally

more than n votes, the voter's vote can be deduced.

3.3.2 Ballot secrecy in small polling stations

The difficulties of large scale corruption may prohibit our attack in the majority of polling stations, however, our attack is feasible in small polling stations found in rural districts. For example, let us consider the 2007 legislative elections in the district of Toul [Est07]. This district has 75350 eligible voters registered at 193 polling stations. Accordingly, the average polling station has 390 registered voters, but the variance is large. Indeed, 33 polling stations have between 50 and 99 voters, 9 polling stations have less then 50 voters, and the smallest two polling stations have 8, respectively 16, voters. Moreover, the attack is simplified by non-participating voters. In these small polling stations it is thus sufficient to corrupt a small number of voters to reveal a voter's vote, furthermore, the final outcome of the election would not change as it is based on 75350 eligible voters.

4 Solution: Ballot independence

Our attacks exploit the possibility of replaying a voter's ballot, or a variant of the voter's ballot, without detection, and can be attributed to the lack of ballot independence in Helios. This section sketches some possible solutions to ensure ballot independence.

4.1 Ballot weeding

The ballots cast by the adversary in our attacks can all be identified. Accordingly, we propose a solution which identifies and rejects such ballots. First, we assume that the signature of knowledge scheme (Section 2.2) is revised to ensure non-malleability, in particular, this will ensure that the response components of signatures cannot be changed. Secondly, we assume that the decryption algorithm will only decrypt ciphertexts (a, b), where $a, b \in \mathbb{Z}_p^*$. Finally, the election officer should reject any ballot that contains a ciphertext that already exists on the bulletin board (this check can be performed in Step 4 of the protocol execution, see Section 2.3). Witness that our first constraint eliminates the attacks described in Sections 3.2.1 & 3.2.5, the second eliminates the attacks described in Sections 3.2.3 & 3.2.4, and the final constraint eliminates those in Sections 3.1 & 3.2.2 (the attack described in Section 3.2.3 can also be eliminated by the final constraint). This solution is simple and can easily be implemented in a future version of Helios.

4.2 Binding ballots to voters

Ballot weeding requires additional cryptographic assumptions and a special mechanism to reject ballots meaningfully related to those already present on

the bulletin board. In this section, we propose techniques to prevent the construction of meaningfully related ballots that will be accepted by the bulletin board, namely, we bind the link between a voter and her ballot; it follows that any meaningfully related ballot constructed by the adversary will be rejected by the bulletin board because the ballot is not bound to the adversary.

Unique identifiers. Based upon inspiration from Gennaro [Gen95, §4.2], Cramer, Gennaro & Schoenmakers [CGS97], and Damgård, Jurik & Nielsen [DJ01, DJN10], we use unique identifiers to ensure that signatures of knowledge are associated with distinct voters. This is achieved by including the voter's identity in the challenges used by signatures of knowledge. More precisely, given a voter's identity id, the sign algorithm (Section 2.2) is modified as follows: on input (a, b), such that $a \equiv g^r \bmod p$ and $b \equiv h^r \cdot g^m \bmod p$, let challenge $c_m = \mathcal{H}(a_{\mathsf{min}}, b_{\mathsf{min}}, \ldots, a_{\mathsf{max}}, b_{\mathsf{max}}, id) - \sum_{i \in \{\mathsf{min}, \ldots, m-1, m+1, \ldots, \mathsf{max}\}} c_i \pmod q$, where values $a_{\mathsf{min}}, b_{\mathsf{min}}, \ldots, a_{\mathsf{max}}, b_{\mathsf{max}}$ and $c_1, \ldots, c_{m-1}, c_{m+1}, \ldots, c_m$ are defined as before. For correctness, the verification algorithm must also be modified, in particular, for candidate signatures constructed by the voter with identity id, the verifier should check $\mathcal{H}(a_{\mathsf{min}}, b_{\mathsf{min}}, \ldots, a_{\mathsf{max}}, b_{\mathsf{max}}, id) \equiv \sum_{\mathsf{min} \leq i \leq \mathsf{max}} c_i \pmod q$.

Eligibility verifiability. The electronic voting protocol proposed by Juels, Catalano & Jakobsson [JCJ05] – which has been implemented by Clarkson, Chong & Myers [CCM08, CCM07] as Civitas – requires ballots to be bound to private voter credentials. This provides *eligibility verifiability* [KRS10]: anyone can check that each ballot published on the bulletin board was cast by a registered voter and at most one ballot is tallied per voter. It is likely that eligibility verifiability enforces ballot independence, but the provision of eligibility verifiability appears to be expensive, in particular, Juels, Catalano & Jakobsson and Clarkson, Chong & Myers assume the existence of an infrastructure for voter credentials.

4.3 Critique of our solutions

Our *ballot weeding* solution is particularly attractive because it adheres to Benaloh's notion of *ballot casting assurance* [Ben06, Ben07] which asserts that the ballot encryption device (the browser script in this instance) does not know the voter's identity. The ballot casting assurance principle is important because knowledge of the voter's identity could be used to infer the likelihood of auditing and this information can be used to influence the behaviour of the ballot encryption device, in particular, if a ballot is unlikely to be audited, then the device may act maliciously, for example, by encrypting a different vote. By comparison, the *unique identifiers* solution would necessarily require that the voter's identity be revealed to the ballot encryption device. Moreover, extending Helios to provide eligibility verifiability would require a considerable extension to the Helios code-base. Accordingly, we adopt the ballot weeding solution and,

in the next section, we show that this is sufficient to ensure ballot secrecy, in the formal setting.

5 Formal proof of ballot secrecy

In this section, we formally prove that our solution is sufficient for ballot secrecy using the applied pi calculus [AF01, RS11].

5.1 Applied pi calculus

Let us recall the applied pi calculus. We assume an infinite set of *names* $a, b, c, \ldots, k, \ldots, m, n, \ldots, s, \ldots$, an infinite set of *variables* $x, y, z, \ldots$, and a *signature* Σ consisting of a finite set of *function symbols*, each with an associated arity. We use metavariables u, w to range over both names and variables. *Terms* L, M, N, T, U, V are built by applying function symbols to names, variables, and other terms. We write $\{^M/_x\}$ for the *substitution* that replaces the variable x with the term M. Arbitrarily large substitutions can be written as $\{^{M_1}/_{x_1}, \ldots, ^{M_l}/_{x_l}\}$ and the letters σ and τ range over substitutions. We write $N\sigma$ for the result of applying σ to the free variables of term N. A term is *ground* when it does not contain variables.

The signature Σ is equipped with an *equational theory* E, that is, a set of equations of the form $M = N$, where the terms M, N are defined over the signature Σ. We define equality modulo the equational theory, written $=_E$, as the smallest equivalence relation on terms that contains E and is closed under application of function symbols, substitution of terms for variables and bijective renaming of names. We write $M =_E N$ when the equation $M = N$ is in the theory E, and keep the signature implicit. When E is clear from its usage, we may abbreviate $M =_E N$ as $M = N$. The negation of $M =_E N$ is denoted $M \neq_E N$ (and similarly abbreviated $M \neq N$).

Processes and *extended processes* are defined in the usual way (Figure 2). We write $\nu\, \tilde{u}$ for the (possibly empty) series of pairwise-distinct binders $\nu\, u_1. \cdots .\nu\, u_l$. The active substitution $\{^M/_x\}$ can replace the variable x for the term M in every process it comes into contact with and this behaviour can be controlled by restriction, in particular, the process $\nu\, x.(\{^M/_x\} \mid P)$ corresponds exactly to let $x = M$ in P. Arbitrarily large active substitutions can be obtained by parallel composition and we occasionally abbreviate $\{^{M_1}/_{x_1}\} \mid \ldots \mid \{^{M_l}/_{x_l}\}$ as $\{^{M_1}/_{x_1}, \ldots, ^{M_l}/_{x_l}\}$ or $\{^{\tilde{M}}/_{\tilde{x}}\}$. We also use σ and τ to range over active substitutions, and write $N\sigma$ for the result of applying σ to the free variables of N. Extended processes must have at most one active substitution for each variable and there is exactly one when the variable is under restriction. The only minor change compared to [AF01] is that conditional branches now depend on formulae $\phi, \psi ::= M = N \mid M \neq N \mid \phi \wedge \psi$. If M and N are ground, we define $[\![M = N]\!]$ to be true if $M =_E N$ and false otherwise. The semantics of $[\![\ ]\!]$ is then extended to formulae in the standard way.

Figure 2 Syntax for processes

$P, Q, R ::=$	(plain) processes
0	null process
$P \mid Q$	parallel composition
$!P$	replication
$\nu\, n.P$	name restriction
if ϕ then P else Q	conditional
$u(x).P$	message input
$\overline{u}\langle M\rangle.P$	message output
$A, B, C ::=$	extended processes
P	plain process
$A \mid B$	parallel composition
$\nu\, n.A$	name restriction
$\nu\, x.A$	variable restriction
$\{^M/_x\}$	active substitution

The *scope* of names and variables are delimited by binders $u(x)$ and $\nu\, u$. The set of bound names is written $\mathrm{bn}(A)$ and the set of bound variables is written $\mathrm{bv}(A)$; similarly we define the set of free names $\mathrm{fn}(A)$ and free variables $\mathrm{fv}(A)$. Occasionally, we write $\mathrm{fn}(M)$ (and $\mathrm{fv}(M)$ respectively) for the set of names (and respectively variables) which appear in term M. An extended process is *closed* when every variable x is either bound or defined by an active substitution.

We define a *context* $C[\_]$ to be an extended process with a hole. We obtain $C[A]$ as the result of filling $C[\_]$'s hole with the extended process A. An *evaluation context* is a context whose hole is not in the scope of a replication, a conditional, an input, or an output. A context $C[\_]$ closes A when $C[A]$ is closed.

A *frame*, denoted φ or ψ, is an extended process built from the null process 0 and active substitutions $\{^M/_x\}$, which are composed by parallel composition and restriction. The *domain* $\mathrm{dom}(\varphi)$ of a frame φ is the set of variables that φ exports, that is, the set of variables x for which φ contains an active substitution $\{^M/_x\}$ such that x is not under restriction. Every extended process A can be mapped to a frame $\varphi(A)$ by replacing every plain process in A with 0.

5.1.1 Operational semantics

The operational semantics are defined by three relations: *structural equivalence* ($\equiv$), *internal reduction* ($\rightarrow$), and *labelled reduction* ($\xrightarrow{\alpha}$). These relations satisfy the rules in Figure 3 and are defined such that: structural equivalence is the smallest equivalence relation on extended processes that is closed by α-conversion of both bound names and bound variables, and closed under application of evaluation contexts; internal reduction is the smallest relation on extended processes closed under structural equivalence and application of eval-

Figure 3 Semantics for processes

$$
\begin{array}{lrcl}
\textsc{Par-0} & A & \equiv & A \mid 0 \\
\textsc{Par-A} & A \mid (B \mid C) & \equiv & (A \mid B) \mid C \\
\textsc{Par-C} & A \mid B & \equiv & B \mid A \\
\textsc{Repl} & !P & \equiv & P \mid !P
\end{array}
$$

$$
\begin{array}{lrcl}
\textsc{New-0} & \nu\, n.0 & \equiv & 0 \\
\textsc{New-C} & \nu\, u.\nu\, w.A & \equiv & \nu\, w.\nu\, u.A \\
\textsc{New-Par} & A \mid \nu\, u.B & \equiv & \nu\, u.(A \mid B) \\
 & & & \text{where } u \notin \mathrm{fv}(A) \cup \mathrm{fn}(A)
\end{array}
$$

$$
\begin{array}{lrcl}
\textsc{Alias} & \nu\, x.\{^M/_x\} & \equiv & 0 \\
\textsc{Subst} & \{^M/_x\} \mid A & \equiv & \{^M/_x\} \mid A\{^M/_x\} \\
\textsc{Rewrite} & \{^M/_x\} & \equiv & \{^N/_x\} \\
 & & & \text{where } M =_E N
\end{array}
$$

$$
\begin{array}{ll}
\textsc{Comm} & \overline{c}\langle x\rangle.P \mid c(x).Q \to P \mid Q \\
\textsc{Then} & \text{if } \phi \text{ then } P \text{ else } Q \to P \quad \text{if } [\![\phi]\!] = \mathsf{true} \\
\textsc{Else} & \text{if } \phi \text{ then } P \text{ else } Q \to Q \quad \text{otherwise}
\end{array}
$$

$$\textsc{In} \quad c(x).P \xrightarrow{c(M)} P\{^M/_x\}$$

$$\textsc{Out-Atom} \quad \overline{c}\langle u\rangle.P \xrightarrow{\overline{c}\langle u\rangle} P$$

$$\textsc{Open-Atom} \quad \frac{A \xrightarrow{\overline{c}\langle u\rangle} A' \qquad u \neq c}{\nu\, u.A \xrightarrow{\nu\, u.\overline{c}\langle u\rangle} A'}$$

$$\textsc{Scope} \quad \frac{A \xrightarrow{\alpha} A' \qquad u \text{ does not occur in } \alpha}{\nu\, u.A \xrightarrow{\alpha} \nu\, u.A'}$$

$$\textsc{Par} \quad \frac{A \xrightarrow{\alpha} A' \qquad \mathrm{bv}(\alpha) \cap \mathrm{fv}(B) = \mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset}{A \mid B \xrightarrow{\alpha} A' \mid B}$$

$$\textsc{Struct} \quad \frac{A \equiv B \qquad B \xrightarrow{\alpha} B' \qquad B' \equiv A'}{A \xrightarrow{\alpha} A'}$$

uation contexts; and for labelled reductions α is a *label* of the form $c(M)$, $\overline{c}\langle u\rangle$, or $\nu\, u.\overline{c}\langle u\rangle$ such that u is either a channel name or a variable of base type.

5.1.2 Equivalence

The definition of observational equivalence [AF01] quantifies over all contexts which makes proofs difficult, therefore we adopt labelled bisimilarity in this article. Labelled bisimilarity relies on an equivalence relation between frames, called static equivalence.

Definition 1 (Static equivalence)**.** *Two closed frames φ and ψ are statically equivalent, denoted $\varphi \approx_s \psi$, if $\mathrm{dom}(\varphi) = \mathrm{dom}(\psi)$ and there exists a set of names $\tilde{n}$ and substitutions σ, τ such that $\varphi \equiv \nu\, \tilde{n}.\sigma$ and $\psi \equiv \nu\, \tilde{n}.\tau$ and for all terms M, N such that $\tilde{n} \cap (\mathrm{fn}(M) \cup \mathrm{fn}(N)) = \emptyset$, we have $M\sigma =_E N\sigma$ holds if and only if $M\tau =_E N\tau$ holds. Two closed extended processes A, B are statically equivalent, written $A \approx_s B$, if their frames are statically equivalent; that is, $\varphi(A) \approx_s \varphi(B)$.*

The relation $\approx_s$ is called *static* equivalence because it only examines the current state of the processes, and not the processes' dynamic behaviour. The following definition of labelled bisimilarity captures the dynamic part.

Definition 2 (Labelled bisimilarity)**.** *Labelled bisimilarity ($\approx_l$) is the largest symmetric relation $\mathcal{R}$ on closed extended processes such that $A\, \mathcal{R}\, B$ implies:*

1. *$A \approx_s B$;*
2. *if $A \to A'$, then $B \to^* B'$ and $A'\, \mathcal{R}\, B'$ for some B';*
3. *if $A \xrightarrow{\alpha} A'$ such that $\mathrm{fv}(\alpha) \subseteq \mathrm{dom}(A)$ and $\mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset$, then $B \to^* \xrightarrow{\alpha} \to^* B'$ and $A'\, \mathcal{R}\, B'$ for some B'.*

Definitions of observational equivalence and labelled bisimilarity have been shown to coincide [Liu11].

5.2 Modelling Helios in applied pi

We start by constructing a suitable signature Σ to capture the cryptographic primitives used by Helios and define an equational theory E to capture the relationship between these primitives.

5.2.1 Signature

We adopt the following signature.

$$\Sigma = \{\mathsf{ok}, \mathsf{zero}, \mathsf{one}, \bot, \mathsf{fst}, \mathsf{snd}, \mathsf{pair}, *, +, \circ, \mathsf{partial}, \mathsf{checkspk}, \mathsf{penc}, \mathsf{spk}, \mathsf{dec}\}$$

Functions ok, zero, one, $\bot$ are constants; fst, snd are unary functions; dec, pair, partial, $*$, $+$, $\circ$ are binary functions; checkspk, penc are ternary functions; and spk is a function of arity four. We adopt infix notation for $*$, $+$, and $\circ$.

The term $\mathsf{penc}(T, N, M)$ denotes the encryption of plaintext M, using random nonce N and key T. The term $U * U'$ denotes the homomorphic combination of ciphertexts U and U', the corresponding operation on plaintexts is written

$M + M'$ and $N \circ N'$ on nonces. The partial decryption of ciphertext U using key L is denoted $\mathsf{partial}(L, U)$. The term $\mathsf{spk}(T, N, M, U)$ represents a signature of knowledge that proves U is a ciphertext under the public key T on the plaintext M using nonce N and such that M is either the constant zero or one. We introduce tuples using pairings and, for convenience, we occasionally abbreviated $\mathsf{pair}(M_1, \mathsf{pair}(\ldots, \mathsf{pair}(M_n, \bot)))$ as $(\!|M_1, \ldots, M_n|\!)$, and $\mathsf{fst}(\mathsf{snd}^{i-1}(M))$ is denoted $\pi_i(M)$, where $i \in \mathbb{N}$. We use the equational theory E that asserts functions $+, *, \circ$ are commutative and associative, and includes the equations:

$$\mathsf{fst}(\mathsf{pair}(x, y)) = x \tag{E1}$$

$$\mathsf{snd}(\mathsf{pair}(x, y)) = y \tag{E2}$$

$$\mathsf{zero} + \mathsf{one} = \mathsf{one} \tag{E3}$$

$$\mathsf{zero} + \mathsf{zero} = \mathsf{zero} \tag{E4}$$

$$\mathsf{dec}(x_{\mathsf{sk}}, \mathsf{penc}(\mathsf{pk}(x_{\mathsf{sk}}), x_{\mathsf{rand}}, x_{\mathsf{plain}})) = x_{\mathsf{plain}} \tag{E5}$$

$$\mathsf{dec}(\mathsf{partial}(x_{\mathsf{sk}}, \mathit{ciph}), \mathit{ciph}) = x_{\mathsf{plain}} \tag{E6}$$
$$\text{where } \mathit{ciph} = \mathsf{penc}(\mathsf{pk}(x_{\mathsf{sk}}), x_{\mathsf{rand}}, x_{\mathsf{plain}})$$

$$\mathsf{penc}(x_{\mathsf{pk}}, y_{\mathsf{rand}}, y_{\mathsf{plain}}) * \mathsf{penc}(x_{\mathsf{pk}}, z_{\mathsf{rand}}, z_{\mathsf{plain}}) \tag{E7}$$
$$= \mathsf{penc}(x_{\mathsf{pk}}, y_{\mathsf{rand}} \circ z_{\mathsf{rand}}, y_{\mathsf{plain}} + z_{\mathsf{plain}})$$

$$\mathsf{checkspk}(x_{\mathsf{pk}}, \mathit{ball}, \mathsf{spk}(x_{\mathsf{pk}}, x_{\mathsf{rand}}, \mathsf{zero}, \mathit{ball})) = \mathsf{ok} \tag{E8}$$
$$\text{where } \mathit{ball} = \mathsf{penc}(x_{\mathsf{pk}}, x_{\mathsf{rand}}, \mathsf{zero})$$

$$\mathsf{checkspk}(x_{\mathsf{pk}}, \mathit{ball}, \mathsf{spk}(x_{\mathsf{pk}}, x_{\mathsf{rand}}, \mathsf{one}, \mathit{ball})) = \mathsf{ok} \tag{E9}$$
$$\text{where } \mathit{ball} = \mathsf{penc}(x_{\mathsf{pk}}, x_{\mathsf{rand}}, \mathsf{one})$$

Equation E6 allows plaintext M to be recovered from ciphertext $\mathsf{penc}(\mathsf{pk}(L), N, M)$ given partial decryption $\mathsf{partial}(L, \mathsf{penc}(\mathsf{pk}(L), N, M))$, when the partial decryption is constructed using the private key L. Equation E7 represents the homomorphic combination of ciphertexts. The Equations E8 and E9 allow the verification of signatures of knowledge $\mathsf{spk}(T, N, M, \mathsf{penc}(T, N, M))$, when $M \in \{\mathsf{zero}, \mathsf{one}\}$. The remaining equations are standard.

Example 1. *Given randomness N, N', plaintexts $(M, M') \in \{(\mathsf{zero}, \mathsf{zero}), (\mathsf{zero}, \mathsf{one}), (\mathsf{one}, \mathsf{zero})\}$, and public key T, one can construct a signature of knowledge $L = \mathsf{spk}(T, N \circ N', M + M', \mathsf{penc}(T, N, M) * \mathsf{penc}(T, N', M'))$. Then* checkspk *applied to the public key T, the homomorphically combined ciphertexts $\mathsf{penc}(T, N, M) * \mathsf{penc}(T, N', M')$, and the signature L is equal to* ok *using Equations E3, E7, E8, and E9*

5.2.2 Helios process specification

In the applied pi calculus, it is sufficient to model the parts of the voting system which need to be trusted for ballot secrecy; all the remaining parts of the system are controlled by the adversarial environment. Accordingly, we assume the

existence of at least two honest voters $\mathcal{A}$, $\mathcal{B}$; since this avoids the scenario where ballot secrecy of an individual voter is compromised by collusion amongst all the remaining voters. In addition, the following trust assumptions are required.

- At least one trustee is honest
- The election officer runs the bulletin board honestly:
  - Voters $\mathcal{A}$, $\mathcal{B}$ have authentic channels with the bulletin board
  - Signatures of knowledge are checked*
  - Ballots which contain a ciphertext that already exists on the bulletin board are rejected*
  - The tally is correctly computed*
  - The trustees have an authentic channel with the bulletin board
- The browser script is trusted and has the correct public key of the election

(Assumptions marked with * could be performed by an honest trustee, rather than the bulletin board.) Although neither voters nor observers can verify that there exists an honest trustee, an assurance of trust is provided by distribution. The necessity to trust the election officer to run the bulletin board is less desirable and work-in-progress [PAM10] aims to weaken this assumption; moreover, to further distribute trust assumptions, the trustees could also check signatures and tallying. Finally, trust in the browser script can be obtained by using software written by a reputable source or writing your own code.

The trusted components are modelled by the administration process $A^{\phi}_{\ell,n}$ and voting process V_ℓ defined in Figure 4. For generality, the voting process V_ℓ is parametrised by the number of candidates ℓ. Similarly, the administration process $A^{\phi}_{\ell,n}$ is parametrised by the number of candidates ℓ, the number of voters n, and a formula ϕ; the formula ϕ defines the checks performed by the bulletin board before accepting a ballot. We will consider several variants of Helios (including the original Helios 2.0 protocol and our fixed scheme) by considering suitable formula that we call *Helios process specifications*.

Definition 3 (Helios process specification)**.** *A formula $\phi_{\ell,\bar{n}}$ is a* Helios process specification*, if* $\mathrm{fv}(\phi_{\ell,\bar{n}}) \subseteq \{y_1, \ldots, y_{\bar{n}}, y_{\mathsf{ballot}}, z_{\mathsf{pk}}\}$.

The voting process V_ℓ contains free variables $x_1^{\mathsf{vote}}, \ldots, x_\ell^{\mathsf{vote}}$ to represent the voter's vote (which is expected to be encoded using constants zero and one) and the free variable x_{auth} represents the channel shared by the voter and the bulletin board. The definition of the process V_ℓ corresponds to the description of the browser script (Figure 1). The administration process $A^{\phi}_{\ell,n}$ is parametrised by the number of candidates ℓ, the number of voters n, and a Helios process specification ϕ. The restricted name sk_T models the tallier's secret key and the public part $\mathsf{pk}(sk_T)$ is included in the process's frame. The restricted names a_1 and a_2 model authentic channels between the two honest voters and the bulletin board, and the channel name d captures the authentic channel with the honest

Figure 4 Helios process specification

Let ℓ be some number of candidates, $n \geq 2$ be some number of voters, and ϕ be a Helios process specification. The administration process $A^{\phi}_{\ell,n}$ and voting process V_ℓ are defined below.

$$
\begin{array}{lcl}
V_\ell & = & \nu\, r_1\ . \\
 & & \mathsf{let}\ ciph_1 = \mathsf{penc}(z_{\mathsf{pk}}, r_1, x_1^{\mathsf{vote}})\ \mathsf{in} \\
 & & \mathsf{let}\ spk_1 = \mathsf{spk}(z_{\mathsf{pk}}, r_1, x_1^{\mathsf{vote}}, ciph_1))\ \mathsf{in} \\
 & & \vdots \\
 & & \nu\, r_\ell\ . \\
 & & \mathsf{let}\ ciph_\ell = \mathsf{penc}(z_{\mathsf{pk}}, r_\ell, x_\ell^{\mathsf{vote}})\ \mathsf{in} \\
 & & \mathsf{let}\ spk_\ell = \mathsf{spk}(z_{\mathsf{pk}}, r_\ell, x_\ell^{\mathsf{vote}}, ciph_\ell))\ \mathsf{in} \\
 & & \mathsf{let}\ \hat{r} = r_1 \circ \cdots \circ r_\ell\ \mathsf{in} \\
 & & \mathsf{let}\ \widehat{ciph} = ciph_1 * \cdots * ciph_\ell\ \mathsf{in} \\
 & & \mathsf{let}\ \widehat{vote} = x_1^{\mathsf{vote}} + \cdots + x_\ell^{\mathsf{vote}}\ \mathsf{in} \\
 & & \mathsf{let}\ \widehat{spk} = \mathsf{spk}(z_{\mathsf{pk}}, \hat{r}, \widehat{vote}, \widehat{ciph})\ \mathsf{in} \\
 & & \overline{x_{\mathsf{auth}}}\langle (\!| ciph_1, \ldots, ciph_\ell,\ spk_1, \ldots, spk_\ell,\ \widehat{spk} |\!) \rangle
\end{array}
$$

$$
A^{\phi}_{\ell,n} \quad = \quad \nu\, sk_T, a_1, a_2, d\ .\ (\_ \mid BB^{\phi}_{\ell,n} \mid T_\ell \mid \{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\})
$$

$$
\begin{array}{lcl}
BB^{\phi}_{\ell,n} & = & a_1(y_1)\ .\ \overline{c}\langle y_1 \rangle\ .\ a_2(y_2)\ .\ \overline{c}\langle y_2 \rangle\ . \\
 & & a_3(y_3)\ .\ \mathsf{if}\ \phi_{\ell,2}\{^{y_3}/_{y_{\mathsf{ballot}}}\}\ \mathsf{then} \\
 & & \cdots\ a_n(y_n)\ .\ \mathsf{if}\ \phi_{\ell,n-1}\{^{y_n}/_{y_{\mathsf{ballot}}}\}\ \mathsf{then} \\
 & & \mathsf{let}\ tally_1 = \pi_1(y_1) * \cdots * \pi_1(y_n)\ \mathsf{in} \\
 & & \cdots \mathsf{let}\ tally_\ell = \pi_\ell(y_1) * \cdots * \pi_\ell(y_n)\ \mathsf{in} \\
 & & \overline{d}\langle (\!| tally_1, \ldots, tally_\ell |\!) \rangle\ . \\
 & & d(y_{\mathsf{partial}})\ . \\
 & & \overline{c}\langle y_{\mathsf{partial}} \rangle\ . \\
 & & \overline{c}\langle (\!| \mathsf{dec}(\pi_1(y_{\mathsf{partial}}), tally_1), \ldots, \mathsf{dec}(\pi_\ell(y_{\mathsf{partial}}), tally_\ell) |\!) \rangle
\end{array}
$$

$$
\begin{array}{lcl}
T_\ell & = & d(y_{\mathsf{tally}})\ . \\
 & & \overline{d}\langle (\!| \mathsf{partial}(sk_T, \pi_1(y_{\mathsf{tally}})), \ldots, \mathsf{partial}(sk_T, \pi_\ell(y_{\mathsf{tally}})) |\!) \rangle
\end{array}
$$

trustee. To ensure the adversary has access to messages sent on private channels, communication is relayed on the public channel c. The sub-process $BB^{\phi}_{\ell,n}$ represents the bulletin board and T_ℓ represents the tallier. The bulletin board accepts ballots from each voter and checks they are valid using the Helios process specification ϕ (this predicate will be discussed in more detail below). Once all ballots have been submitted, the bulletin board homomorphically combines the ciphertexts and sends the encrypted tallies to the tallier for decryption. (The necessity for all voters to participate is included for simplicity, in particular, our bulletin board does not weed ballots containing invalid proofs.) The tallier

receives the homomorphic combinations of ballots y_{tally} and derives a partial decryption for each candidate; these partial decryptions are sent to the bulletin board and the election result is published.

The voting process V_ℓ is parametrised by a substitution σ, where variables $x_1^{\mathsf{vote}}, \ldots, x_\ell^{\mathsf{vote}} \in \mathrm{dom}(\sigma)$; these variables must be parametrised to encode a vote for at most one candidate, that is, there exists at most one integer $i \in \{1, \ldots, \ell\}$ such that $\Sigma \vdash x_i^{\mathsf{vote}}\sigma = \mathsf{one}$. Formally, we define valid parametrisations using the notion of *candidate substitutions*.

Definition 4 (Candidate substitution)**.** *Given some number of candidates ℓ and a substitution σ, we say σ is a* candidate substitution *if*

$$\Sigma \vdash (x_1^{\mathsf{vote}} + \cdots + x_\ell^{\mathsf{vote}})\sigma = \mathsf{zero} \vee \Sigma \vdash (x_1^{\mathsf{vote}} + \cdots + x_\ell^{\mathsf{vote}})\sigma = \mathsf{one}$$

It follows immediately that bitstrings $m_1, \ldots, m_\ell$ generated during Step 2 of Figure 1 can be modelled as candidate substitutions.

The application of our model is demonstrated in the following example.

Example 2. *Let ℓ be some number of candidates, $n \geq 2$ be some number of voters, and ϕ be a Helios process specification. An election with voters $\mathcal{A}$ and $\mathcal{B}$ who select candidate substitutions σ and τ, and such that the other $n-2$ voters are controlled by the adversary, can be modelled by the process $A^{\phi}_{\ell,n}[V_\ell\{^{a_1}/_{x_{\mathsf{auth}}}\}\sigma \mid V_\ell\{^{a_2}/_{x_{\mathsf{auth}}}\}\tau]$.*

Ballot validity. In Helios 2.0, the election officer considers a ballot to be valid if the signature proofs of knowledge hold. Accordingly, we can model the Helios administration by the process $A^{\phi^{\mathsf{orig}}}_{\ell,n}$ where the Helios process specification ϕ^{orig}, parametrised by the number of candidates ℓ, is defined as follows.

$$\begin{aligned}\phi_\ell^{\mathsf{orig}} \triangleq\ & \mathsf{checkspk}(z_{\mathsf{pk}}, \pi_1(y_{\mathsf{ballot}}) * \cdots * \pi_\ell(y_{\mathsf{ballot}}), \pi_{2\cdot\ell+1}(y_{\mathsf{ballot}})) = \mathsf{ok} \wedge \\ & \mathsf{checkspk}(z_{\mathsf{pk}}, \pi_1(y_{\mathsf{ballot}}), \pi_{\ell+1}(y_{\mathsf{ballot}})) = \mathsf{ok} \wedge \ldots \wedge \\ & \mathsf{checkspk}(z_{\mathsf{pk}}, \pi_\ell(y_{\mathsf{ballot}}), \pi_{2\cdot\ell}(y_{\mathsf{ballot}})) = \mathsf{ok}\end{aligned}$$

We have shown that these checks are insufficient to ensure ballot secrecy (Section 3). Our ballot weeding solution, proposed in Section 4.1, additionally requires that the ciphertexts inside the ballot do not appear on the bulletin board. This revised scheme can be modelled using the Helios process specification ϕ^{sol}, parametrised by the number of candidates ℓ and number of ballots already on the bulletin board $\bar{n}$, defined as follows.

$$\phi^{\mathsf{sol}}_{\ell,\bar{n}} \triangleq \phi^{\mathsf{orig}}_\ell \wedge \pi_{2\cdot\ell+2}(y_{\mathsf{ballot}}) = \bot \wedge \bigwedge_{\substack{i,j\in\{1,\ldots,\ell\},\\ k\in\{1,\ldots,\bar{n}\}}} \pi_i(y_k) \neq \pi_j(y_{\mathsf{ballot}})$$

We can also model a naïve solution that would consist in weeding only identical ballots by considering the Helios process specification ϕ^{ident}, parametrised by

the number of candidates ℓ and number of ballots already on the bulletin board $\bar{n}$, defined below.

$$\phi^{\mathsf{ident}}_{\ell,\bar{n}} \triangleq \phi^{\mathsf{orig}}_{\ell} \wedge \pi_6(y_{\mathsf{ballot}}) = \bot \wedge \bigwedge_{k \in \{1,\ldots,\bar{n}\}} y_{\mathsf{ballot}} \neq y_k$$

We have already shown that removing exact duplicates is insufficient because it would fail to detect variants of our attack whereby the contents of a ballot are permuted. In the next section, we formally show that Helios 2.0 (modelled using ϕ^{orig}) and the naïve solution (modelled using ϕ^{ident}) do not satisfy ballot secrecy, and that our proposed solution (modelled using ϕ^{sol}) does satisfy ballot secrecy.

5.3 Formal analysis: Ballot secrecy

Based upon [KR05, DKR06, DKR09], and as previous discussed (see *related work* in Section 1), we formalise ballot secrecy for two voters $\mathcal{A}$ and $\mathcal{B}$ with the assertion that an adversary cannot distinguish between a situation in which voter $\mathcal{A}$ votes for candidate t and voter $\mathcal{B}$ votes for candidate t', from another situation in which $\mathcal{A}$ votes t' and $\mathcal{B}$ votes t. Formally, this is captured by Definition 5.

Definition 5 (Ballot secrecy)**.** *Given a Helios process specification ϕ, we say* ballot secrecy is satisfied *if for all integers $\ell \in \mathbb{N}^*$ and $n \geq 2$, and for all candidates substitutions σ and τ, we have*

$$A^{\phi}_{\ell,n}[V_\ell\{{}^{a_1}/_{x_{\mathsf{auth}}}\}\sigma \mid V_\ell\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\tau] \approx_l A^{\phi}_{\ell,n}[V_\ell\{{}^{a_1}/_{x_{\mathsf{auth}}}\}\tau \mid V_\ell\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma]$$

The ballot secrecy definition proposed by Delaune, Kremer & Ryan considered a vote to be an arbitrary name, whereas a vote in our setting must be a series of the constant symbols zero and one, such that their combination by application of the function $+$ is also a constant zero and one; it follows that Definition 5 is a straightforward variant of the original.

The Helios 2.0 protocol does not satisfy our privacy definition (Lemma 3) and naïve weeding solutions are also insufficient (Lemma 4).

Lemma 3. *The Helios process specification ϕ^{orig} does not satisfy ballot secrecy.*

Intuitively, the proof of Lemma 3 is due to the environment's ability to replay $\mathcal{A}$'s ballot, therefore introducing an observable difference: the result will include two instances of $\mathcal{A}$'s vote. Formally, this follows immediately from the proof Lemma 4.

Lemma 4. *The Helios process specification ϕ^{ident} does not satisfy ballot secrecy.*

Proof. Consider $\ell = 2$, $n = 3$, $\sigma = \{{}^{\mathsf{zero}}/_{x_1^{\mathsf{vote}}}, {}^{\mathsf{one}}/_{x_2^{\mathsf{vote}}}\}$ and $\tau = \{{}^{\mathsf{one}}/_{x_1^{\mathsf{vote}}}, {}^{\mathsf{zero}}/_{x_2^{\mathsf{vote}}}\}$. We consider a sequence of transitions where the two voters output their ballots and then the adversary chooses its ballots to be a permutation of the

first voter's ballot. Namely, if the first voter's ballot is $(\!|ciph, ciph', spk, spk', \widehat{spk}|\!)$ then the adversary outputs $(\!|ciph', ciph, spk', spk, \widehat{spk}|\!)$. Formally, this corresponds to the following transitions

$$A^{\phi}_{\ell,n}[V_\ell\{^{a_1}/_{x_{\mathsf{auth}}}\}\sigma \mid V_\ell\{^{a_2}/_{x_{\mathsf{auth}}}\}\tau] \\ \to \xrightarrow{\nu\, x.\overline{c}\langle x\rangle} \to \xrightarrow{\nu\, y.\overline{c}\langle y\rangle} \xrightarrow{c((\!|\pi_2(x),\pi_1(x),\pi_4(x),\pi_3(x),\pi_5(x)|\!))} \to^* \xrightarrow{\nu\, z.\overline{c}\langle z\rangle} \nu\, \tilde{n}.\tau_1$$

for some names $\tilde{n}$ and substitution τ_1, such that:

$$\mathsf{dec}(\pi_1(z), \pi_1(x) * \pi_1(y) * \pi_2(x))\tau_1 =_E \mathsf{one} + \mathsf{one}$$

This labelled transition has to matched by

$$A^{\phi}_{\ell,n}[V_\ell\{^{a_1}/_{x_{\mathsf{auth}}}\}\tau \mid V_\ell\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma] \\ \to \xrightarrow{\nu\, x.\overline{c}\langle x\rangle} \to \xrightarrow{\nu\, y.\overline{c}\langle y\rangle} \xrightarrow{c((\!|\pi_2(x),\pi_1(x),\pi_4(x),\pi_3(x),\pi_5(x)|\!))} \to^* \xrightarrow{\nu\, z.\overline{c}\langle z\rangle} \nu\, \tilde{n}.\tau_2$$

for some names $\tilde{n}$ and substitution τ_2, such that:

$$\mathsf{dec}(\pi_1(z), \pi_1(x) * \pi_1(y) * \pi_2(x))\tau_2 =_E \mathsf{one}$$

It follows immediately that $\nu\, \tilde{n}.\tau_1 \not\approx_s \nu\, \tilde{n}.\tau_2$ and, hence, ϕ^{ident} does not satisfy ballot secrecy. □

In contrast, removing duplicates up to permutation ensures ballot secrecy.

Theorem 1. *The Helios process specification ϕ^{sol} satisfies ballot secrecy.*

ProVerif is an automatic tool that can check equivalence in the applied pi calculus [BAF08]. Although ProVerif has been successfully used to prove ballot secrecy (for example, in the Fujioka, Okamoto & Ohta protocol [DRS08]), it cannot prove Theorem 1, at the time of writing, for two main reasons. Firstly, ProVerif cannot prove equivalences under the homomorphic equation (Equation E7). Secondly, our theorem states ballot secrecy for any number n of participants and ProVerif cannot handle parametrised processes (see Paiola & Blanchet [PB12, PB11, Pai10] for some initial progress in this direction). We proceed by constructing a relation that relates $A^{\phi}_{\ell,n}[V_\ell\{^{a_1}/_{x_{\mathsf{auth}}}\}\sigma \mid V_\ell\{^{a_2}/_{x_{\mathsf{auth}}}\}\tau]$ and $A^{\phi}_{\ell,n}[V_\ell\{^{a_1}/_{x_{\mathsf{auth}}}\}\tau \mid V_\ell\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma]$, and all their successors, such that it satisfies the three properties of Definition 2. In particular, the two final frames (containing the result of the election) should be statically equivalent.

Definition 6 (Valid ballot)**.** *A term T is said to be a* valid ballot *in an election with ℓ candidates if $[\![\phi^{\mathsf{sol}}_{\ell,0}\{^{T}/_{y_{\mathsf{ballot}}}\}]\!] = \mathsf{true}$.*

By definition, the bulletin board accepts only valid ballots. A key step to proving static equivalence is to show that any valid ballot submitted to the bulletin board by the environment is "equivalent" to a term of the form $(\!|\mathsf{penc}(z_{\mathsf{pk}}, N_1, M_1), \ldots, \mathsf{penc}(z_{\mathsf{pk}}, N_\ell, M_\ell), S_1, \ldots, S_{\ell+1}|\!)$, where $\{^{M_1}/_{x^{\mathsf{vote}}_1}, \ldots, ^{M_\ell}/_{x^{\mathsf{vote}}_\ell}\}$ is a candidate substitution. This allows us to deduce that the election outcome produced by $A^{\phi}_{\ell,n}[V_\ell\{^{a_1}/_{x_{\mathsf{auth}}}\}\sigma \mid V_\ell\{^{a_2}/_{x_{\mathsf{auth}}}\}\tau]$ is exactly the same as in $A^{\phi}_{\ell,n}[V_\ell\{^{a_1}/_{x_{\mathsf{auth}}}\}\tau \mid$

$V_\ell\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma]$. We can then conclude the proof of Theorem 1 by showing that the partial decryptions and the encrypted ballots of honest voters do not leak any extra information to the adversary. The full proof appears in Appendix B.

5.4 Limitations

The limitations of our model, which we introduced to simplify the presentation and proof, are detailed below; we believe a full security proof should follow using similar reasoning. We make use of a (standard) definition of ballot secrecy which is limited to elections with two honest voters [KR05, DKR06, DKR09]. In addition, the definition of ballot secrecy does not consider parallel composition of protocol executions and we therefore recommend using distinct keys for each election (although we believe it should be sufficient to include an election identifier – for example, the election fingerprint – in the challenge hashes included within signatures of knowledge, similar to the methodology in Section 4.2). The administrative process $A^{\phi}_{\ell,n}$ enforces an ordering on voters (namely, the voter using private channel a_1 must vote first, followed by the voter using private channel a_2, and then any remaining voters – controlled by the adversarial environment – can vote); this limitation could be overcome by parametrising $A^{\phi}_{\ell,n}$ with the channel names to restrict and by a minor unification of the bulletin process $BB^{\phi}_{\ell,n}$, however, this generalisation is of limited interest and would come at the cost of over-complicating the proof. In addition, the administrative process $A^{\phi}_{\ell,n}$ does not permit revoting. The signature and equational theory do not capture low-level technical details of public keys, in particular, we consider a single honest tallier and we do not model distributed keys nor signatures of knowledge to verify correct construction of both keys and partial decryptions (nonetheless, we include Equation E6 which models decryption of a ciphertext using a correctly constructed partial decryption). Finally, we offer the usual caveat to formal analysis and acknowledge that our result does not imply the absence of real-world attacks (see, for example, [RS98, AR00, AR02, War03, War05]). It may, therefore, be possible to modify the ballot in a way that would not be captured by our analysis. We partly overcome these limitations in our further work [BCP+11] by presenting a variant of Helios that is provably secure in a cryptographic setting.

6 Attacks against other schemes

This section demonstrates that the absence of ballot independence can be exploited in other electronic voting protocols to violate privacy. In particular, we demonstrate replay attacks against schemes by Sako & Kilian [SK94] and Schoenmakers [Sch99], and we show that the malleable cryptographic scheme adopted by Lee *et al.* [LBD+04] can be exploited to launch attacks.

6.1 Exploiting replays in the protocol by Sako & Kilian

The Sako & Kilian [SK94] electronic voting scheme capitalises upon advances in cryptography to improve the Banaloh & Yung protocol [BY86]. The scheme is interesting because it was one of the first electronic voting protocols to adopt the Fiat-Shamir heuristic to derive non-interactive proofs (this evolution was key for the development of end-to-end verifiable electronic voting systems): a three-round zero-knowledge proof consisting of a commitment, challenge and response can be reduced to a non-interactive proof by replacing the challenge with a hash on the commitment. However, we show that the application of the Fiat-Shamir heuristic compromises ballot secrecy. In particular, the interactive nature of zero-knowledge proofs guarantees freshness, whereas, non-interactive proofs, derived using the Fiat-Shamir heuristic, do not assure freshness. This can be exploited by a replay attack to violate ballot secrecy.

6.1.1 Protocol description

The scheme is based upon a pair of *partially compatible homomorphic encryption* functions, that is, a pair of functions f_1, f_2 over $\mathbb{Z}_q$, where q is prime, such that for all $i, j \in \{1, 2\}$ the following properties are satisfied:

- $f_i(x + y) = f_i(x) \cdot f_i(y)$, where $x, y \in \mathbb{Z}_q$
- Distributions $(f_i(x), f_j(y))$ and $(f_i(x), f_j(x))$ are computationally indistinguishable, where x and y are chosen uniformly in $\mathbb{Z}_q$.

The voting protocol is defined for $m \in \mathbb{N}$ voters as follows.

Setup. Talliers $\mathcal{T}$ and $\mathcal{T}'$ publish public keys k and k' for a public key encryption scheme E (which need not be homomorphic).

Voting. Given vote $v_i \in \{-1, 1\}$, the voter generates nonces $x_i, x_i' \in \mathbb{Z}_q$ such that $v_i = x_i + x_i'$ and constructs her ballot as follows:

$$\begin{array}{rcl} Y_i & = & f_1(x_i) \\ Y_i' & = & f_2(x_i') \\ Z_i & = & E(k, x_i) \\ Z_i' & = & E(k', x_i') \end{array}$$

In addition, the voter is required to prove $x_i + x_i' \in \{1, -1\}$ in zero-knowledge. However, to avoid an interactive proof, the Fiat-Shamir heuristic is applied to derive a signature of knowledge σ_i. (For brevity we omit the construction of σ_i, see [SK94, Figure 1] for details.)

Tallying. Given ballots $Y_1, Y_1', Z_1, Z_1', \sigma_1, \ldots, Y_n, Y_n', Z_n, Z_n', \sigma_n$, tallier $\mathcal{T}$ decrypts each Z_i to recover $\hat{x}_i$ and checks $Y_i = f_1(\hat{x}_i)$, similarly, tallier $\mathcal{T}'$ decrypts Z_i' to recover $\hat{x}_i'$ and checks $Y_i' = f_1(\hat{x}_i')$; the talliers also check the signature of knowledge σ_i. The talliers publish $V = \sum_{i=1}^{m} \hat{x}_i$ and $V' = \sum_{i=1}^{m} \hat{x}_i'$, and the

result is $T = V + V'$, which can be verified by checking $f_1(V) = \prod_{i=1}^{m} Y_i$ and $f_2(V') = \prod_{i=1}^{m} Y'_i$.

6.1.2 Attacking ballot secrecy

We show that the voting protocol by Sako & Kilian does not satisfy ballot secrecy, by presenting a replay attack which allows an adversary to reveal a voter's vote. Intuitively, an adversary may observe the ballot posted by a particular voter and recast this ballot by corrupting dishonest voters. The multiple occurrences of the voter's ballot will leak information in the tally and the adversary can exploit this knowledge to violate the voter's privacy. An informal description of the attack will now be presented in the case of three eligible voters.

Let us consider an election with three eligible voters who have identities id_1, id_2 and id_3. Suppose that voters id_1, id_2 are honest and id_3 is a dishonest voter controlled by the adversary. Further assume that the adversary has observed the ballot

$$Y_k, Y'_k, Z_k, Z'_k, \sigma_k$$

being cast by the voter whose privacy will be compromised.

Replaying a ballot. As shown by Gennaro [Gen95], an adversary can replay the ballot $Y_k, Y'_k, Z_k, Z'_k, \sigma_k$, thereby violating ballot independence. (The violation of ballot independence is due to the adversary's ability to cast the *same* vote as the honest voter.) Since the ballot was constructed by an honest voter, it is trivial to see that it will be considered valid by the talliers. We will now show how the lack of ballot independence can be exploited to violate privacy.

Violating privacy. The bulletin board will be constructed as follows

$$Y_1, Y'_1, Z_1, Z'_1, \sigma_1, Y_2, Y'_2, Z_2, Z'_2, \sigma_2, Y_k, Y'_k, Z_k, Z'_k, \sigma_k, V, V'$$

where $k \in \{1, 2\}$, $V = x_1 + x_2 + x_k$ and $V' = x'_1 + x'_2 + x'_k$. It follows from the protocol description that $v_i = x_i + x'_i$, where $i \in \{1, 2, k\}$, and the result $T = V + V' = v_1 + v_2 + v_k$. Since there will be at least two votes for the candidate voter id_k voted for, the voter's vote can be revealed: if $T \geq 2$, then $v_k = 1$; otherwise $v_k = -1$. It follows that the voter's privacy has been compromised; moreover, the vote of the remaining honest voter is $T - 2 \cdot v_k$.

6.1.3 Independence and the Fiat-Shamir heuristic

The interactive nature of zero-knowledge proofs guarantees freshness, because every proof contains a unique challenge, and this ensures independence. By comparison, non-interactive proofs, derived using the Fiat-Shamir heuristic, do not assure freshness, in particular, non-interactive proofs can be replayed. As a consequence, application of the Fiat-Shamir heuristic may compromise the security of cryptographic protocols and we have shown how application of the heuristic erodes privacy in the electronic voting scheme by Sako & Kilian. This

demonstrates that the use of the Fiat-Shamir heuristic requires some care and highlights the necessity for thorough security analysis.

6.1.4 Generalising replay attacks

The replay attack against Helios, and the voting protocol by Sako & Kilian, can be generalised to other schemes where an adversary can observe a ballot cast by a particular voter and replay this ballot verbatim. In particular, the voting protocol by Schoenmakers [Sch99] fits this description.

Exploiting replays in the protocol by Schoenmakers. The electronic voting protocol by Schoenmakers [Sch99] is based upon [CFSY96, CGS97]. The scheme explicitly aims to provide efficient small-scale elections (for example, boardroom elections) and, given that our attack is particularly well suited to small-scale elections, we find it interesting to study the possibility of violating ballot secrecy in this setting. Ballot independence is not provided [Sch99, §5] and we exploit privacy using a replay attack. The attack description is straightforward and follows immediately from our discussion; accordingly, we omit the details and refer the interested reader to our technical report [SC11, §3].

6.1.5 Possible solutions: Weeding duplicate ballots

Our attacks against the voting protocols by Sako & Kilian and Schoenmakers exploit the possibility of replaying a voter's ballot without detection. We believe it should be sufficient for the election officer to reject any duplicate ballots to ensure ballot secrecy, alternatively, the *unique identifiers* solution (Section 4.2) may also be suitable. Proving the security of these solutions remains an open problem.

6.2 Exploiting malleability in the protocol by Lee *et al.*

The Lee *et al.* [LBD+04] electronic voting scheme adopts an offline tamper-resistant hardware device to ensure receipt freeness; more precisely, the hardware device takes an ElGamal encrypted vote as input and outputs a re-encrypted ciphertext, this prevents a voter proving how she voted by reconstruction as she does not know the nonce introduced for re-encryption. In addition, the hardware device provides a Designated Verifier Proof of re-encryption, thereby allowing the voter to verify that the device behaved correctly. The device is assumed to be offline and, hence, communication between the voter and the device is assumed to be untappable.

6.2.1 Background: Multiplicative homomorphic ElGamal

The scheme uses multiplicative homomorphic ElGamal, rather than the additive variant presented in Section 2.1. The operations for key generation, homomorphic combination and re-encryption are standard; albeit, the result of

homomorphic combination is the multiplication of plaintexts, rather than the addition of plaintexts. We recall the operations for encryption and decryption below.

Encryption. Given a message m and a public key h, select a random nonce $r \in_R \mathbb{Z}_q^*$ and derive the ciphertext $(a, b) = (g^r \bmod p,\ m \cdot h^r \bmod p)$.

Distributed decryption. Given a ciphertext (a, b), each trustee $i \in n$ computes the partial decryption $k_i = a^{x_i}$. The plaintext $m = b/(k_1 \cdot \ldots \cdot k_n) \bmod p$.

The application of these primitives to derive the scheme by Lee *et al.* will be discussed in the next section.

6.2.2 Protocol description

An election is created by naming an election officer, selecting a set of mixers, and choosing a set of trustees. The trustees generate a distributed public key pair and the election officer publishes the public key on the bulletin board. (For robustness, threshold ElGamal may be used; we omit these details for brevity.) The election officer also publishes the candidate list, the public keys of eligible voters, and the public keys of the tamper-resistant hardware devices. Informally, the steps that the participants take during an election are as follows.

1. The voter constructs an ElGamal ciphertext (a, b) containing her vote v and sends the ciphertext to her tamper-resistant hardware device.

2. The hardware device re-encrypts the voter's ciphertext to produce (a', b') and computes a Designated Verifier Proof of re-encryption τ. The device also derives a signature σ on the re-encryption. The hardware device returns $(a', b'), \sigma, \tau$ to the voter.

3. If the signature and proof are valid, then the voter generates a signature σ' on the message σ using her private key. The voter submits her ballot $(a', b'), \sigma, \sigma'$ to the bulletin board.

4. Individual voters can check that their ballots appear on the bulletin board and can be assured that the ciphertext (a', b') contains their vote v by verifying the Designated Verifier Proof τ.

5. Voters and observers can check that ballots were cast by registered voters by verifying signatures σ', and are assured that each voter cast at most one ballot by checking that no voter signed two values. In addition, voters and observers should verify signatures σ for receipt freeness.

6. After some predefined deadline, valid ballots (that is, ballots associated with valid signatures σ and σ') are submitted to the mixers. Anyone can check that mixing is performed correctly.

7. Each of the trustees publishes a partial decryption for every ciphertext output by the mix. Anyone can verify these proofs.

8. The election officer decrypts each ciphertext and publishes the election result. Anyone can check these decryptions.

See Lee *et al.* [LBD+04] for further details.

6.2.3 Attacking ballot secrecy

We show that the voting protocol by Lee *et al.* [LBD+04] does not satisfy ballot secrecy by recalling the attack by Dreier, Lafourcade & Lakhnech [DLL11] that exploits malleability to reveal a voter's vote. Intuitively, an adversary may identify a voter's encrypted vote on the bulletin board, since it is signed by the voter. This ciphertext can be submitted to a tamper-resistant hardware device (possibly after re-encryption) and the device will return $(\hat{a}, \hat{b}), \hat{\sigma}, \hat{\tau}$; the ballot $(\hat{a}, \hat{b}), \hat{\sigma}, \hat{\sigma}'$ can then be submitted by the adversary to the bulletin board, where $\hat{\sigma}'$ is a signature on $\hat{\sigma}$ constructed by a registered voter under the adversary's control. As explained in Section 6.1.2, the multiple occurrences of the voter's ballot will leak information in the tally and the adversary can exploit this knowledge to violate the voter's privacy.

Variant exploiting homomorphic encryption. The adversary can exploit the homomorphic properties of ElGamal to avoid casting the *same* vote as an honest voter. In this variant, suppose the adversary wants to recover the vote from ballot $(a'_k, b'_k), \sigma_k, \sigma'_k$, the adversary derives the ciphertext $(c, d) = (a'_k, b'_k) \cdot (c', d')$, where (c', d') is an ElGamal ciphertext containing some message m. The adversary submits the ciphertext (c, d) to a tamper-resistant hardware device and the device will return $(\hat{c}, \hat{d}), \hat{\sigma}, \hat{\tau}$; the ballot $(\hat{c}, \hat{d}), \hat{\sigma}, \hat{\sigma}'$ can be submitted by the adversary to the bulletin board, where $\hat{\sigma}'$ is a signature on $\hat{\sigma}$ constructed by a registered voter. The output of the mix will include the adversaries re-encrypted ciphertext and the election officer will publish $m \cdot v$ on the bulletin board, where ciphertext (a'_k, b'_k) includes the vote v. This variant of the attack is interesting because the adversary's ballots are undetectable, in particular, weeding ballots would clearly not be sufficient to ensure privacy.

6.2.4 Exploiting replays in protocols based upon mixnets

In homomorphic election schemes, voters encrypt their votes and ciphertexts are homomorphically combined before decryption. By comparison, in mixnet election schemes, voters encrypt their votes, ciphertexts are shuffled, and individual ciphertexts are decrypted. (The decryption of individual ciphertexts in mixnet elections can provide an advantage over homomorphic elections, since ballot construction can be simplified, for example, the number of ciphertexts used by Helios corresponds to the number of candidates, whereas, one ciphertext is sufficient in mixnet elections.) Privacy is ensured in homomorphic election

schemes by never revealing the contents of individual ciphertexts, whereas, privacy is ensured in mixnet election schemes by breaking the link between the mix's input and output. As highlighted by Pfitzmann & Pfitzmann [PP89], independence is necessary in mixnets because the input ciphertexts are eventually decrypted, therefore, any meaningfully related ciphertexts input to the mixnet can be meaningfully related once the mix's output is decrypted. In the context of electronic voting, it follows that a voter's privacy can be violated if an adversary can construct a ciphertext meaningfully related to the voter's ciphertext and the election result contains exactly two votes satisfying the adversary's relation.

7 Relationships between security properties

The variants of our attack (Section 3.2) abuse ballot malleability to violate privacy and our ballot weeding solution achieves privacy by ensuring ballots are independent. In this section, we study the relationships between independence and privacy, and independence and malleability.

7.1 Independence and privacy are unrelated properties

In the context of Helios, we have shown that ballot independence is sufficient for ballot secrecy, however, we will now present examples that suggest independence and privacy are unrelated in a more general context.

A protocol with independence but no privacy. Consider a variant of the fixed Helios voting scheme in which each of the trustees publish a partial decryption of individual ciphertexts (rather than a partial decryption of the homomorphically combined ciphertexts, that is, the encrypted tally). Intuitively, this variant preserves ballot independence but does not satisfy ballot secrecy, since the partial decryptions allow votes to be recovered from ballots and the link between a voter and her ballot is known. Formally, this variant is captured by modelling the Helios administrator process as $\bar{A}_n^{\phi^{\mathsf{sol}}}$, defined in Figure 5. The violation of ballot secrecy can be witnessed since

$$\bar{A}_2^{\phi^{\mathsf{sol}}}[V\{{}^{a_1}/_{x_{\mathsf{auth}}}\}\sigma \mid V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\tau] \not\approx_l \bar{A}_2^{\phi^{\mathsf{sol}}}[V\{{}^{a_1}/_{x_{\mathsf{auth}}}\}\tau \mid V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma]$$

where $\sigma = \{{}^{\mathsf{zero}}/_{x_1^{\mathsf{vote}}}, {}^{\mathsf{one}}/_{x_2^{\mathsf{vote}}}\}$ and $\tau = \{{}^{\mathsf{one}}/_{x_1^{\mathsf{vote}}}, {}^{\mathsf{zero}}/_{x_2^{\mathsf{vote}}}\}$. Similarly, a further variant of the fixed Helios scheme in which each of the trustees publishes their private key at the end of the voting phase, rather than a partial decryption of the encrypted tally, also satisfies independence but not ballot secrecy.

A protocol with privacy but no independence. Consider a voting scheme in which each voter broadcasts their vote on an anonymous communication channel. Formally, the voter is modelled by the process $P = \overline{c}\langle x_{\mathsf{vote}}\rangle$, where variable x_{vote} is parametrised by the voter's vote. For ballot secrecy it is sufficient

Figure 5 Helios administrator that preserves independence but not privacy

Given the number of voters $n \geq 2$ the administration process $\bar{A}_n^{\phi^{\mathsf{sol}}}$ is defined below, where process T is presented in Figure 4.

$$\bar{A}_n^{\phi^{\mathsf{sol}}} = \nu\, sk_T, a_1, a_2, d\,.\,(\_ \mid \overline{BB}_n^{\phi^{\mathsf{sol}}} \mid !T \mid \{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\})$$

$$\begin{aligned}\overline{BB}_n^{\phi^{\mathsf{sol}}} = {} & a_1(y_1)\,.\,\overline{c}\langle y_1\rangle\,.\,a_2(y_2)\,.\,\overline{c}\langle y_2\rangle\,.\\ & a_3(y_3)\,.\,\text{if } \phi^{\mathsf{sol}}\{^{y_3}/_{y_{\mathsf{ballot}}}\} \text{ then}\\ & \cdots\; a_n(y_n)\,.\,\text{if } \phi^{\mathsf{sol}}\{^{y_n}/_{y_{\mathsf{ballot}}}\} \text{ then}\\ & \overline{d}\langle(\!|\pi_1(y_1), \pi_2(y_1)|\!)\rangle\,.\,d(z_1)\,.\,\overline{c}\langle z_1\rangle\,.\\ & \cdots\,.\,\overline{d}\langle(\!|\pi_1(y_n), \pi_2(y_n)|\!)\rangle\,.\,d(z_n)\,.\,\overline{c}\langle z_n\rangle\end{aligned}$$

to show $P\{^M/_{x_{\mathsf{vote}}}\} \mid P\{^N/_{x_{\mathsf{vote}}}\} \approx_l P\{^N/_{x_{\mathsf{vote}}}\} \mid P\{^M/_{x_{\mathsf{vote}}}\}$ for all ground terms M and N; this result trivially holds by structural equivalence and hence the scheme satisfies ballot secrecy. However, independence is intuitively violated in this setting, because an adversary may observe the voting system and replay a previously cast vote, that is, an adversary can cast the same vote as another voter (without knowing which voter). In addition, it follows that early results are available in this scheme.

We also expect some published electronic voting schemes based upon blind signatures to satisfy ballot secrecy but not independence; in particular, a more realistic example of a protocol that satisfies this property is the protocol by Fujioka, Okamoto & Ohta [FOO92] under the assumption that duplicates are not rejected. Indeed, independence can be violated by a verbatim replay of the signed committed vote.

Nonetheless, we believe a weaker property exists: *privacy and authenticated ballots implies independence*, where the term *authenticated ballot* means the link between an arbitrary ballot and associated voter is known (for example, our unique identifiers solution uses authenticated ballots). Informally, this can be witnessed as follows: suppose a system satisfies privacy and authenticated ballots but not independence, it follows that an adversary can identify a voter's ballot and, since there is no independence, replay that ballot; privacy is then violated, as we have shown in this article, hence deriving a contradiction. In addition, Bernhard, Pereira & Warinschi [BPW12] propose a context where privacy implies independence.

7.2 Non-malleability is stronger than independence

Non-malleability asserts that an adversary can only construct *meaningfully related* ballots if the related ballots are constructed by the adversary [DDN91, BDPR98, DDN00]. By comparison, given an election's bulletin board, ballot independence asserts that an adversary can only construct a ballot which will be accepted by the bulletin board and be meaningfully related to an existing

ballot on the board, if the adversary constructed both ballots. It intuitively follows that non-malleability implies ballot independence, since an adversary that is unable to construct a ballot meaningfully related to a non-adversarial ballot, is also unable to construct a ballot that will be accepted by an election's bulletin board *and* be meaningfully related to a non-adversarial ballot from the bulletin board. By contrast, our ballot weeding solution (Section 4.1) suggests that non-malleability is not necessary for independence. Indeed, we have shown that an adversary can form several ballots that are meaningfully related to an initial one. All ballots are of the right format (in particular they contain a valid proof). In that sense, ballots are malleable. However, the additional checks of the ballot box (weeding duplicates) will reject them. Our unique identifier solution (Section 4.2) provides further evidence to support our claim that non-malleability is not necessary for independence. In this setting, a non-adversarial ballot from the bulletin board can be manipulated using all of the techniques defined in Section 3.2 to derive a meaningfully related ballot, nevertheless, if the adversary constructs such a ballot, then the ballot will be rejected by the bulletin board, because it is not bound to the adversary's identity. Our examples therefore provide evidence to suggest that non-malleability is not necessary for independence.

7.3 Discussion

In this article, we cannot make any definitive mathematical statements about the relationships between independence and privacy or independence and non-malleability, because independence has not been formally defined. Nonetheless, we hope this section provides some insight into the relationships we expect.

8 Related work

The attack against Helios that we discover relies upon the lack of ballot independence. The concept of independence was introduced by Chor *et al.* [CGMA85] and the possibility of compromising security properties due to the lack of independence has been considered, for example, in [CR87, PP89, DDN91, DDN00, Gen00]. In the context of electronic voting, Gennaro [Gen95] demonstrates that the application of the Fiat-Shamir heuristic in the Sako-Kilian electronic voting protocol [SK94] violates ballot independence, and Wikström [Wik06, Wik08] studies non-malleability for mixnets to achieve ballot independence. By comparison, we focus on the violation of ballot secrecy rather than fairness, and exploit the absence of ballot independence to compromise privacy. Similar results have been shown against mixnets [Pfi94].

Our attack is also reliant on the voter's ability to cast a ballot as a function of another voter's ballot, for example, our basic attack (Section 3.1) applies the identity function and our variant in Section 3.2.2 performs a permutation on the ballot's internal structure. In related work, Benaloh [Ben96] demonstrates that a simplified version of his voting scheme allows the administrator's private

key to be recovered by an adversary who constructs (and casts) a ballot as a function of other voters' ballots.

Estehghari & Desmedt [ED10] claim to present an attack which undermines privacy and end-to-end verifiability in Helios. However, their attack is dependent on compromising a voter's computer, a vulnerability which is explicitly acknowledged by the Helios specification [AMPQ09]: *"a specifically targeted virus could surreptitiously change a user's vote and mask all of the verifications performed via the same computer to cover its tracks."* Accordingly, [ED10] represents an exploration of known vulnerabilities rather than an attack.

Other studies of Helios have also been conducted, in particular, Langer *et al.* [Lan10, LSBV10] and Volkamer & Grimm [VG10] study privacy in Helios. Langer *et al.* propose a taxonomy of informal privacy requirements [Lan10, LSBV10, LSB$^+$10] to facilitate a more fine-grained comparison of electronic voting systems, this framework is used to analyse Helios and the authors claim ballot secrecy is satisfied if the adversary only has access to public data [Lan10, LSBV10]. Volkamer & Grimm introduce the *k-resilience* metric [VG10, Vol09] to calculate the number of honest participants required for ballot secrecy in particular scenarios, this framework is used to analyse Helios and the authors claim ballot secrecy is satisfied if the software developers are honest and the key holders do not collude [VG10]. Our attacks invalidate these claims. We believe the erroneous results reported by Langer *et al.* are due to a lack of formally written proofs, and the approach by Volkamer & Grimm failed because only some particular scenarios were considered.

In our further work with Bernhard *et al.* [BCP$^+$11], we present a computational security proof demonstrating that any variant of Helios using an IND-CCA2 secure encryption scheme provides ballot secrecy and, more concretely, propose a variant using the Naor-Yung paradigm [NY90] to derive an IND-CCA2 secure encryption scheme from ElGamal. In this setting, independence is achieved using non-malleable ballots. Intuitively, the use of ElGamal and a suitable signature of knowledge scheme allows us to derive an IND-CCA2 secure encryption scheme; indeed, Tsiounis & Yung [TY98] and Schnorr & Jakobsson [SJ00] provide some evidence to support this hypothesis, however, these results are presented in the generic group model and proving this result under weaker assumptions is an open problem [SG98, SG02]. Nonetheless, it appears that a more efficient provably secure variant of Helios can be derived.

In principle, work by Bernhard, Pereira & Warinschi [BPW12] supports the aforementioned proposal that a more efficient variant of Helios exists: Bernhard, Pereira & Warinschi prove that an IND-CPA encryption scheme and a suitable signature of knowledge can be combined to derive NM-CPA security, and the *minivoting scheme* [BCP$^+$11] is shown to satisfy ballot secrecy for any NM-CPA secure encryption scheme. Bernhard, Pereira & Warinschi argue that the minivoting scheme forms the basis of Helios and informally claim that Helios is therefore secure since the transformation from minivoting to Helios does not affect ballot secrecy. However, the minivoting scheme is restricted to ballots containing a single ciphertext, hence the security of Helios can only be guaranteed for ballots that contain a vote for a single candidate (for example, in

referendums).

A further variant of Helios is proposed by Bulens, Giry & Pereira [BGP11] using mixnets rather than homomorphic encryption. As highlighted by Pfitzmann & Pfitzmann [PP89], independence is necessary in mixnets because the input ciphertexts are eventually decrypted and Bulens, Giry & Pereira use an IND-CCA2 secure encryption scheme to derive non-malleability and therefore independence.

Delaune, Kremer & Ryan [DKR06, DKR09] have shown that a variant of the Lee *et al.* protocol satisfies coercion resistance for two honest voters; but, based upon our preliminary results [CS11], Dreier, Lafourcade & Lakhnech [DLL11] demonstrate an attack against privacy for three voters, when one voter is under the adversary's control[3]. Furthermore, using a stronger definition of coercion resistance, Küsters & Truderung [KT09] have demonstrated a forced abstention attack; in addition, Küsters & Truderung propose a variant of the scheme by Lee *et al.* which is claimed to satisfy their stronger definition. In this article, we show a new attack against the original Lee *et al.* protocol and show that the revised scheme by Küsters & Truderung might not be secure under reasonable assumptions.

A preliminary version of this work [CS11] appeared at the 24th Computer Security Foundations Symposium. By comparison, in this article, we provide further variants of our attack (a preliminary presentation of these variants appears on ePrint [Smy12]), a more detailed description of our results, a generalisation of our ballot secrecy proof to a setting with arbitrary many candidates, and include complete proofs. In addition, we show that other electronic voting protocols are vulnerable to our attack and we discuss the relationships between independence and privacy, and independence and malleability.

9 Conclusion

This article identifies a vulnerability in the Helios 2.0 electronic voting protocol which can be used to violate ballot secrecy. Critics may argue that an attack is unrealistic due its high cost; indeed, in some cases, the attack may change the outcome of an election (that is, the votes introduced for the purposes of violating privacy may swing the result), and large scale privacy invasions would be expensive in terms of the required number of dishonest voters. However, if the views of these critics are to be entertained, then we must revise the standard definitions of ballot secrecy in the literature – for example, [KR05, DKR06, BHM08] – because Helios cannot satisfy them. Furthermore, we believe all voters should be considered equally and, hence, the preservation of ballot secrecy should be universal. But, for elections using Helios, our case study demonstrates the contrary: in French legislative elections a coalition of voters can gain some information about a voter's vote in an arbitrary polling station and, moreover, if the number

[3]The formal model by Dreier, Lafourcade & Lakhnech includes the voter's signature on the signed re-encrypted ciphertext and this is exploited by their attack; by comparison, the model by Delaune, Kremer & Ryan omits this detail and therefore the attack cannot be witnessed.

of voters registered at a particular polling station is small (for example, in a rural setting), then a voter's privacy can be violated by a few dishonest voters. It follows that privacy of individual voters can be compromised by a few dishonest voters and, accordingly, we believe our attack is significant. To address the problem, we have introduced a variant of the Helios protocol which has been shown to satisfy definitions of ballot secrecy in the applied pi calculus and in our further work [BCP+11] we present a security proof in the cryptographic setting (Section 8 summarises this result). We have also shown that the absence of ballot independence can be similarly exploited in other electronic voting protocols to violate privacy; in particular, we demonstrate verbatim replay attacks against the schemes by Sako & Kilian [SK94] and Schoenmakers [Sch99], and we show that the malleable cryptographic scheme adopted by Lee *et al.* [LBD+04] can be exploited to replay a voter's ballot or a variant of it, thereby violating ballot secrecy. In addition, we argue that independence and privacy are unrelated in general, and non-malleability is strictly stronger than independence. Finally, with the exception of Schoenmakers, all of the vulnerabilities in this article have been acknowledged by the respective protocol authors, in particular, Adida & Pereira have acknowledged the vulnerability in Helios [Adi10, AP10], but since the vulnerability can only be exploited in elections where voters are willing to forfeit their vote to compromise another voter's privacy, they believe an attack would be *"without serious practical impact."* Nonetheless, Adida & Pereira have scheduled a fix for future Helios releases (at the time of writing, the software implementation of Helios has been patched to prevent the replay attack described in Section 3.1, but the software is still vulnerable to the variants described in Section 3.2).

Acknowledgements

We are grateful to Ben Adida, David Bernhard, Christian Cachin, Jeremy Clark, Olivier Pereira, Mark D. Ryan, and the anonymous reviewers of our work. Ben Adida and Olivier Pereira provided constructive comments; in addition, Ben informed us that Douglas Wikström is the contemporaneous discoverer of the attack described in Section 3.1. Discussion with Mark D. Ryan helped clarify the presentation of this article, David Bernhard and Jeremy Clark gave useful feedback on the variants of our attack described in Sections 3.2.3 & 3.2.4, and Christian Cachin highlighted Josh Benaloh's related work.

A Signatures of knowledge

Helios is reliant on signatures of knowledge to ensure secrecy and integrity of the ElGamal scheme. This appendix presents suitable cryptographic primitives.

A.1 Knowledge of discrete logs

Given the cryptographic parameters (p, q, g) and hash function $\mathcal{H}$ (see Section 2 for details), a signature of knowledge demonstrating knowledge of a discrete logarithm $h = \log_g g^x$ can be derived, and verified, as defined by [CEGP87, CEG88, Sch90]:

Sign. Given x, select a random nonce $w \in_R \mathbb{Z}_q^*$. Compute witness $g' = g^w \bmod p$, challenge $c = \mathcal{H}(g') \bmod q$ and response $s = w + c \cdot x \bmod q$.

Verify. Given h and signature g', s, check $g^s \equiv g' \cdot h^c \pmod p$, where $c = \mathcal{H}(g') \bmod q$.

A valid proof asserts knowledge of x such that $x = \log_g h$, that is, $h \equiv g^x \bmod p$. These proofs are used in distributed ElGamal to ensure secrecy (see Section 2.1).

A.2 Equality between discrete logs

Given the aforementioned cryptographic parameters (p, q, g) and hash function $\mathcal{H}$, a signature of knowledge demonstrating equality between discrete logarithms $\log_f f^x$ and $\log_g g^x$ can be derived, and verified, as defined by [Ped91, CP93]:

Sign. Given f, g, x, select a random nonce $w \in_R \mathbb{Z}_q^*$. Compute witnesses $f' = f^w \bmod p$ and $g' = g^w \bmod p$, challenge $c = \mathcal{H}(f', g') \bmod q$ and response $s = w + c \cdot x \bmod q$.

Verify. Given f, g, h, k and signature f', g', s, check $f^s \equiv f' \cdot h^c \pmod p$ and $g^s \equiv g' \cdot k^c \pmod p$, where $c = \mathcal{H}(f', g') \bmod q$.

A valid proof asserts $\log_f h = \log_g k$, that is, there exists x, such that $h \equiv f^x \bmod p$ and $k \equiv g^x \bmod p$.

Signatures of knowledge demonstrating equality between discrete logarithms are used to ensure integrity of distributed decryption in ElGamal (see Section 2.1), moreover, the signature scheme forms the basis of disjunctive proofs of equality between discrete logs (Section 2.2). Formally, the signature scheme can be used to ensure integrity of distributed decryption in ElGamal as follows. Given a ciphertext (a, b), each trustee would derive a signature on g, a, x_i, where x_i is the trustee's private key share. The ith trustee's signature g_i', a_i', c_i, s_i would be verified with respect to g, a, h_i, k_i, where h_i is the trustee's share of the public key and k_i is the trustee's partial decryption. The signature g_i', a_i', c_i, s_i asserts $\log_g h_i = \log_a k_i$, as required for integrity of decryption.

B Proof of Theorem 1

B.1 Preliminaries

Before commencing our proof, let us first introduce some useful lemmas for the applied pi calculus.

Lemma 5. *Given frames φ, ψ, ground term M and variable $x \notin \mathrm{dom}(\varphi) \cup \mathrm{dom}(\psi)$, we have $\varphi \approx_s \psi$ iff $\varphi \mid \{^M/_x\} \approx_s \psi \mid \{^M/_x\}$.*

Lemma 6. *Given frames φ, ψ, terms M, N, and a variable $x \notin \mathrm{dom}(\varphi) \cup \mathrm{dom}(\psi)$, such that $\varphi = \nu\, \tilde{m}.\sigma$ and $\psi = \nu\, \tilde{n}.\tau$ for some names $\tilde{m}$, $\tilde{n}$ and substitutions σ, τ, we have $\nu\, \tilde{m}.(\sigma \mid \{^M/_x\}) \approx_s \nu\, \tilde{n}.(\tau \mid \{^N/_x\})$ implies $\varphi \approx_s \psi$.*

The proofs of these lemmas are straightforward.

The following lemma shows when static equivalence implies the same branching behaviour for conditionals.

Lemma 7. *Given extended processes $A \equiv C[\text{if } M = N \text{ then } P \text{ else } Q]$ and $B \equiv C'[\text{if } M = N \text{ then } P' \text{ else } Q']$ such that $A \approx_s B$, $(\mathrm{bn}(C) \cup \mathrm{bn}(C')) \cap (\mathrm{fn}(M) \cup \mathrm{fn}(N)) = \emptyset$, $\mathrm{fv}(M) \cup \mathrm{fv}(N) \subseteq \mathrm{dom}(C)$ and $\mathrm{fv}(M) \cup \mathrm{fv}(N) \subseteq \mathrm{dom}(C')$, for some closing evaluation context C, C', terms M, N and processes P, P', Q, Q', then $A \to C[P]$ iff $B \to C'[P']$ and $A \to C[Q]$ iff $B \to C'[Q']$.*

Proof. Suppose $A \equiv C[\text{if } M = N \text{ then } P \text{ else } Q]$ and $B \equiv C'[\text{if } M = N \text{ then } P' \text{ else } Q']$ such that $A \approx_s B$, $(\mathrm{bn}(C) \cup \mathrm{bn}(C')) \cap (\mathrm{fn}(M) \cup \mathrm{fn}(N)) = \emptyset$, $\mathrm{fv}(M) \cup \mathrm{fv}(N) \subseteq \mathrm{dom}(C)$ and $\mathrm{fv}(M) \cup \mathrm{fv}(N) \subseteq \mathrm{dom}(C')$, for some closing evaluation context C, C', terms M, N and processes P, P', Q, Q'. Further suppose $\varphi(C[\text{if } M = N \text{ then } P \text{ else } Q]) = \nu\, \tilde{m}.\sigma$ and $\varphi(C'[\text{if } M = N \text{ then } P' \text{ else } Q']) = \nu\, \tilde{n}.\tau$, for some names $\tilde{m}$ and $\tilde{n}$. By Lemma 8 we have $\nu\, \tilde{m}.\sigma \approx_s \nu\, \tilde{n}.\tau$, because static equivalence is closed under structural equivalence. Moreover, by the definition of static equivalence, for all terms U, V such that $(\tilde{m} \cup \tilde{n}) \cap (\mathrm{fn}(U) \cup \mathrm{fn}(V)) = \emptyset$, we have $U\sigma =_E V\sigma$ iff $U\tau =_E V\tau$.

Let us first show $A \to C[P]$ iff $B \to C'[P']$. For the $\Rightarrow$ implication, suppose $A \to C[P]$. Since $\mathrm{fv}(M) \cup \mathrm{fv}(N) \subseteq \mathrm{dom}(C)$, it must be the case that $M\sigma =_E N\sigma$. We have $\tilde{m} \cup \tilde{n} \subseteq \mathrm{bn}(C) \cup \mathrm{bn}(C')$ by definition of the function φ, and we derive $(\tilde{m} \cup \tilde{n}) \cap (\mathrm{fn}(M) \cup \mathrm{fn}(N)) = \emptyset$ because $(\mathrm{bn}(C) \cup \mathrm{bn}(C')) \cap (\mathrm{fn}(M) \cup \mathrm{fn}(N)) = \emptyset$; it follows that $M\sigma =_E N\sigma$ is a special case of $U\sigma =_E V\sigma$. We derive $M\tau =_E N\tau$ from the implication $(U\sigma =_E V\sigma) \Rightarrow (U\tau =_E V\tau)$. It trivially follows that $B \equiv C'[\text{if } M\tau = M\tau \text{ then } P' \text{ else } Q']$, and by closure of internal reduction under structural equivalence we derive $B \to C'[P']$. The $\Leftarrow$ implications follows by symmetry.

We will now show $A \to C[Q]$ iff $B \to C'[Q']$. For the $\Rightarrow$ implication, suppose $A \to C[Q]$. It must be the case that $M\sigma \neq_E N\sigma$ and, as before, we derive $M\tau \neq_E N\tau$. It trivially follows that $B \equiv C'[\text{if } M\tau = N\tau \text{ then } P' \text{ else } Q']$, and since $\mathrm{fv}(M) \cup \mathrm{fv}(N) \subseteq \mathrm{dom}(C')$ we are assured that terms $M\tau, N\tau$ are ground; by closure of internal reduction under structural equivalence we derive $B \to C'[Q']$. The $\Leftarrow$ implications follows by symmetry. □

This result can naturally be extended to formula. Given ϕ, let us denote the set of free names, respectively variables, in ϕ as $\mathrm{fn}(\phi)$, respectively $\mathrm{fv}(\phi)$.

Corollary 1. *Given extended processes $A \equiv C[\text{if } \phi \text{ then } P \text{ else } Q]$ and $B \equiv C'[\text{if } \phi \text{ then } P' \text{ else } Q']$ such that $A \approx_s B$, $(\mathrm{bn}(C) \cup \mathrm{bn}(C')) \cap \mathrm{fn}(\phi) = \emptyset$, $\mathrm{fv}(\phi) \subseteq \mathrm{dom}(C)$ and $\mathrm{fv}(\phi) \subseteq \mathrm{dom}(C')$, for some closing evaluation context C, C', formulae ϕ and processes P, P', Q, Q', then $A \to C[P]$ iff $B \to C'[P']$ and $A \to C[Q]$ iff $B \to C'[Q']$.*

We conclude this subsection with a useful result stated by Abadi & Fournet [AF01].

Lemma 8. *Static equivalence is closed by structural equivalence.*

B.2 Notations and Definitions

For the remainder of this article, let ℓ be some number of candidates, $n \geq 2$ be some number of voters, and σ and σ' be candidate substitutions.

B.2.1 Notations

We introduce the following notations for all $1 \leq i \leq n$ and $1 \leq j \leq \ell$:

$$
\begin{array}{rcl}
tally_j & = & \pi_j(y_1) * \cdots * \pi_j(y_n) \\
partial_j & = & \mathsf{partial}(sk_T, tally_j) \\
result_j & = & \mathsf{dec}(partial_j, tally_j) \\
ciph_{i,j} & = & \mathsf{penc}(z_{\mathsf{pk}}, r_{i,j}, x_{i,j}^{\mathsf{vote}}) \\
spk_{i,j} & = & \mathsf{spk}(z_{\mathsf{pk}}, r_{i,j}, x_{i,j}^{\mathsf{vote}}, ciph_{i,j}) \\
\widehat{spk_i} & = & \mathsf{spk}(z_{\mathsf{pk}},\ r_{i,1} \circ \cdots \circ r_{i,\ell},\ x_{i,1}^{\mathsf{vote}} + \cdots + x_{i,\ell}^{\mathsf{vote}},\ ciph_{i,1} * \cdots * ciph_{i,\ell}) \\
ballot_i & = & (\!| ciph_{i,1}, \ldots, ciph'_{i,\ell}, spk_{i,1}, \ldots, spk'_{1,\ell}, \widehat{spk_i} |\!) \\
\tau_L & = & \{{}^{M}/_{x_{1,i}^{\mathsf{vote}}} \mid \text{for all } 1 \leq i \leq \ell \text{ such that} \{{}^{M}/_{x_i^{\mathsf{vote}}}\} \in \sigma\} \\
 & & \cup\, \{{}^{N}/_{x_{2,i}^{\mathsf{vote}}} \mid \text{for all } 1 \leq i \leq \ell \text{ such that } \{{}^{N}/_{x_i^{\mathsf{vote}}}\} \in \sigma'\} \\
\tau_R & = & \{{}^{N}/_{x_{1,i}^{\mathsf{vote}}} \mid \text{for all } 1 \leq i \leq \ell \text{ such that} \{{}^{N}/_{x_i^{\mathsf{vote}}}\} \in \sigma'\} \\
 & & \cup\, \{{}^{M}/_{x_{2,i}^{\mathsf{vote}}} \mid \text{for all } 1 \leq i \leq \ell \text{ such that } \{{}^{M}/_{x_i^{\mathsf{vote}}}\} \in \sigma\}
\end{array}
$$

B.2.2 Definitions

Given $N_3, \ldots, N_k$ terms such that $\mathrm{fv}(N_j) \subseteq \{z_{\mathsf{pk}}, y_1, \ldots, y_{j-1}\}$, we define

$$\sigma_{\tilde{N_k}} = \{{}^{ballot_1}/_{y_1}, {}^{ballot_2}/_{y_2}, {}^{N_j}/_{y_j} \mid j \in \{3, \ldots, k\}\}$$

Given an integer $k \in \mathbb{N}^+$ and a term N, we define N^k (resp. $k.N$) to be $N \circ \cdots \circ N$ (resp. $N + \cdots + N$) where N is replicated k times.

We associate to the equational theory E a rewriting system $\mathcal{R}_E$ by orienting the Equations E1,E2 and E5 to E9 from left to right. We denote by E' the

equational theory that asserts functions $+, *, \circ$ are commutative and associative in addition to Equations E3 and E4. $\mathcal{R}_E$ modulo E' forms a convergent rewriting system (modulo E'). We denote by $u \to_E v$ (or often simply $u \to v$) if u modulo E' can be rewritten to v modulo E', using $\mathcal{R}_E$. We denote by $u\downarrow$ a normal form of u modulo E'.

We will say that a term M is *free* w.r.t. a set of names $\tilde{n}$ if it does not contain any name of $\tilde{n}$. We simply say that a term is free when the set of names is clear from the context (typically free w.r.t. to the restricted names of a frame).

B.3 Some useful lemmas

We prove some useful results about our definitions and notations. We first show that ballots accepted by the bulletin board must have a particular form due to the checks performed by $\phi^{\mathsf{sol}}_{\ell,\bar{n}}$.

Lemma 9. *Let ℓ be a number of candidates, $\bar{n} \geq 2$ be an integer, and M be term free w.r.t. $r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}$ and such that $\mathrm{fv}(M) \subseteq \{z_{\mathsf{pk}}, y_1, y_2\}$. Let substitution $\tau \in \{\tau_L, \tau_R\}$ and substitution $\sigma = \{\mathsf{pk}(sk_T)/_{z_{\mathsf{pk}}}, {}^{ballot_1}/_{y_1}, {}^{ballot_2}/_{y_2}\}$. If $[\![\phi^{\mathsf{sol}}_{\ell,\bar{n}}\{{}^{M}/_{y_{\mathsf{ballot}}}\}\sigma\tau]\!] = \mathsf{true}$, then there exists a term*

$$M' = (\!|\mathsf{penc}(z_{\mathsf{pk}}, N_1, M_1), \ldots, \mathsf{penc}(z_{\mathsf{pk}}, N_\ell, M_\ell), S_1, \ldots, S_{\ell+1}|\!)$$

for some terms $M_1, \ldots, M_\ell, N_1, \ldots, N_\ell, S_1, \ldots, S_{\ell+1}$ such that $M\sigma\tau =_E M'\sigma\tau$, M' is free w.r.t. $r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}$, $\mathrm{fv}(M') \subseteq \{z_{\mathsf{pk}}, y_1, y_2\}$, and $\{{}^{M_1}/_{x_1^{\mathsf{vote}}}, \ldots, {}^{M_\ell}/_{x_\ell^{\mathsf{vote}}}\}$ is a candidate substitution.

Proof. Let M, τ and σ be defined as in the Lemma, and suppose $[\![\phi^{\mathsf{sol}}_{\ell,\bar{n}}\{{}^{M}/_{y_{\mathsf{ballot}}}\}\sigma\tau]\!] = \mathsf{true}$. We say that a term N is a *minimal recipe* if it is minimal (in size) among the terms N' such that $N\sigma\tau =_E N'\sigma\tau$. It is easy to check by induction on the size of N that, whenever $N = f(N_1, \ldots, N_k)$ with $f \in \{\mathsf{dec}, \pi_j \mid 1 \leq j \leq \ell\}$ then either $N = \pi_j(x)$ for some j and variable x or $(N\sigma\tau)\downarrow = f((N_1\sigma\tau)\downarrow, \ldots, (N_k\sigma\tau)\downarrow)$ (*).

W.l.o.g. suppose M' is a minimal recipe such that $M\sigma\tau =_E M'\sigma\tau$ and M' is free w.r.t. $r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}$. Further suppose w.l.o.g. that M' is in normal form. We know $[\![\phi^{\mathsf{sol}}_{\ell,\bar{n}}\{{}^{M'}/_{y_{\mathsf{ballot}}}\}\sigma\tau]\!] = \mathsf{true}$. Thus it must be the case that $M'\sigma\tau$ is of the form $(\!|U_1, \ldots, U_\ell, V_1, \ldots, V_\ell, W|\!)$, where for $1 \leq j \leq \ell$ we have $U_j = \mathsf{penc}(\mathsf{pk}(sk_T), R_j, C_j)$, $C_j \in \{\mathsf{zero}, \mathsf{one}\}$ and $\{{}^{C_1}/_{x_1^{\mathsf{vote}}}, \ldots, {}^{C_\ell}/_{x_\ell^{\mathsf{vote}}}\}$ is a candidate substitution. Due to the disequality tests in $\phi^{\mathsf{sol}}_{\ell,\bar{n}}$, it must be the case that M' is of the form $(\!|T_1, \ldots, T_\ell, S_1, \ldots, S_\ell, Z|\!)$ and $T_j \notin \{\pi_k(y_i) \mid 1 \leq k \leq \ell\}$. We have $T_j\sigma\tau = \mathsf{penc}(A_j, B_j, C_j)$. Assume first that $T_j = \pi_k(T'_j)$. Due to (*), we must have T'_j variable, which is excluded by the fact that $T_j \notin \{\pi_k(y_i) \mid 1 \leq k \leq \ell\}$. Thus, due to the equational theory and (*), it must be the case that $T_j = \mathsf{penc}(K_j, N_j, M_j) * \prod_{1\leq k\leq \ell} \pi_k(y_1)^{\alpha_k} * \pi_k(y_2)^{\beta_k}$ where each component is optional and $\alpha_i \in \mathbb{N}$. By convention $\alpha_i = 0$ or $\beta_i = 0$ means that the component is skipped. Assume that one of the α_i or β_i is not null. Then $R_j = r \circ R'_j$ with $r \in \{r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}\}$. Due to the tests in $\phi^{\mathsf{sol}}_{\ell,\bar{n}}$, we know $V_j = \mathsf{spk}(\mathsf{pk}(sk_T), R_j, C_j, V'_j)$.

Let us show that V_j cannot be a signature of knowledge that appears in either $ballot_1\tau_L$ or $ballot_2\tau_L$. Assume (by contradiction) that V_j is a signature of knowledge that appears in either $ballot_1\tau_L$ or $ballot_2\tau_L$. Due to weeding, we cannot have $V_j = \mathsf{spk}_{i,k}$. Indeed, due to the equational theory, this would imply that U_j is equal to a previsouly received cyphertext, which is excluded by weeding. Thus we must have $V_j = \widehat{spk_i}$ for some $i \in \{1, 2\}$. Then $R_j = r_{i,1} \circ \cdots \circ r_{i,\ell}$. In that case, let us have a look at W. We know $W = \mathsf{spk}(\mathsf{pk}(sk_T), R_1 \circ \cdots \circ R_\ell, C_1 + \cdots + C_\ell, U_1 * \cdots * U_\ell)$. Thus W cannot be one of the signatures of knowledge that appear in $ballot_1\tau_L$ or $ballot_2\tau_L$ (the depth of $R_1 \circ \cdots \circ R_\ell$ is too big). Therefore (and due to the equational theory and minimality of Z), we must have $Z = \mathsf{spk}(Z^1, Z^2, Z^3, Z^4)$. Since $r_{i,1} \circ \cdots \circ r_{i,\ell}$ is not deducible, we cannot have $Z^2\sigma\tau =_E r_{i,1} \circ \cdots \circ r_{i,\ell} \circ R_1 \circ R_{j-1} \circ R_{j+1} \circ R_\ell$, contradiction.

We must have $S_j = \mathsf{spk}(S_j^1, S_j^2, S_j^3, S_j^4)$, since V_j cannot be one of the signatures of proof of knowledge that appear in $ballot_1\tau_L$ or $ballot_2\tau_L$, and due to the equational theory. Since r is not deducible, we cannot have $S_j^2\sigma\tau =_E r \circ R'_j$, contradiction. We therefore deduce that $T_j = \mathsf{penc}(K_j, N_j, M_j)$. Moreover, $K_j\sigma\tau =_E \mathsf{pk}(sk_T)$ implies $K_j = z_{\mathsf{pk}}$ and $M_j\sigma\tau =_E \mathsf{zero}$ or one implies $M_j \in \{\mathsf{zero}, \mathsf{one}\}$ due to the equational theory. Due to the validity check, we also deduce that $\{^{M_1}/_{x_1^{\mathsf{vote}}}, \ldots, ^{M_\ell}/_{x_\ell^{\mathsf{vote}}}\}$ is a candidate subsitution. □

Lemma 10. *Let* $\phi_1 = \nu\, sk_T, d, r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}\ .\ (\{^{ballot_1}/_{x_1}\} \mid \{^{ballot_2}/_{x_2}\} \mid \{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\})$. *We have* $\phi_1\tau_L \approx_s \phi_1\tau_R$.

Proof. First, we decompose ϕ_1 and consider $\phi = \nu\, \tilde{n}.\theta$, where $\tilde{n} = \{sk_T, d, r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}\}$ and $\theta = \{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\} \mid \Big\{\{^{ciph_{i,j}}/_{x_{ciph_{i,j}}}\} \mid \{^{spk_{i,j}}/_{x_{spk_{i,j}}}\} \mid \{^{\widehat{spk_i}}/_{x_{\widehat{spk_i}}}\} \;\Big|\; i \in \{1,2\} \wedge 1 \le j \le \ell\Big\}$. It follows immediately that $\phi_1\tau_L \approx_s \phi_1\tau_R$ if and only if $\phi\tau_L \approx_s \phi\tau_R$.

Secondly, witness that the adversary can arbitrarily combine ciphertexts from the frame – namely, ciphertexts $ciph_{1,1}, ciph_{2,1}, \ldots, ciph_{1,\ell}, ciph_{2,\ell}$ – with ciphertexts in the frame or freshly constructed ciphertexts, we enrich the frame ϕ with any such combination of ciphertexts. Formally, for any $\alpha_j, \beta_j \in \mathbb{N}$ and terms P, R we define $C_{\alpha_1,\ldots,\alpha_\ell,\beta_1,\ldots,\beta_\ell,\alpha_4,P,R}$ as follows:

$$\mathsf{penc}(\mathsf{pk}(sk_T), R \circ \textstyle\bigcirc_{1\le j\le \ell} r_{1,j}^{\alpha_j} \circ r_{2,j}^{\beta_j}, P + \sum_{1\le j\le \ell} \alpha_j.x_{1,j}^{\mathsf{vote}} + \beta_j.x_{2,j}^{\mathsf{vote}})$$

We define the extended frame ϕ_e below.

$$\phi_e = \nu\, \tilde{n}.(\theta \mid \{^{C_{\alpha_1,\alpha_2,\alpha_3,\alpha_4,P,R}}/_{x_{\alpha_1,\alpha_2,\alpha_3,\alpha_4,P,R}} \mid \alpha_1,\alpha_2,\alpha_3,\alpha_4 \in \mathbb{N} \text{ and terms } P, R \text{ s.t. } (\mathrm{fn}(P) \cup \mathrm{fn}(R)) \cap \tilde{n} = \emptyset, \mathrm{fv}(P,R) \subseteq \mathrm{dom}(\phi_e) \text{ with no cycle}\})$$

Note that ϕ_e is infinite. By Lemma 6, it is sufficient to show $\phi_e\tau_L \approx_s \phi_e\tau_R$. We introduce the following two claims.

Claim 1. *Let* M *be a term such that* $\mathrm{fv}(M) \cap (\mathrm{fv}(\phi_e) \setminus \mathrm{dom}(\phi_e)) = \emptyset$ *and* $\mathrm{fn}(M) \cap \tilde{n} = \emptyset$. *If* $M\phi_e\tau \to U$ *for some* $\tau \in \{\tau_R, \tau_L\}$, *then there exists* N *such that* $U =_{E'} N\phi_e\tau$ *and* $M\phi_e\tau' \to N\phi_e\tau'$ *for any* $\tau' \in \{\tau_R, \tau_L\}$.

Claim 2. *Let M, N be two terms such that* $(\mathrm{fv}(M) \cup \mathrm{fv}(N)) \cap (\mathrm{fv}(\phi_e) \setminus \mathrm{dom}(\phi_e)) = \emptyset$ *and* $\mathrm{fn}(M, N) \cap \tilde{n} = \emptyset$. *If* $M\phi_e\tau =_{E'} N\phi_e\tau$ *for some* $\tau \in \{\tau_R, \tau_L\}$, *then* $M\phi_e =_{E'} N\phi_e$.

The above claims allow the construction of our proof. Let M, N be two terms such that $\mathrm{fn}(M, N) \cap \tilde{n} = \emptyset$ and $M\phi_e\sigma_{\tilde{N_k}}\tau_L =_E N\phi_e\sigma_{\tilde{N_k}}\tau_L$. We assume (possibly by renaming) that $(\mathrm{fv}(M) \cup \mathrm{fv}(N)) \cap (\mathrm{fv}(\phi_e) \setminus \mathrm{dom}(\phi_e)) = \emptyset$. We have $M\phi_e\tau_L =_E N\phi_e\tau_L$. Thus $(M\phi_e\tau_L)\downarrow =_{E'} (N\phi_e\tau_L)\downarrow$. Applying repeatedly Claim 1, we deduce that there exists M' such that $(M\phi_e\tau_L)\downarrow = M'\phi_e\tau_L$ and $M\phi_e\tau_R \to^* M'\phi_e\tau_R$. Similarly, there exists N' such that $(N\phi_e\tau_L)\downarrow = N'\phi_e\tau_L$ and $N\phi_e\tau_R \to^* N'\phi_e\tau_R$. From $M'\phi_e\tau_L =_{E'} N'\phi_e\tau_L$ and Claim 2, we deduce $M'\phi_e =_{E'} N'\phi_e$. Therefore $M'\phi_e\tau_R =_{E'} N'\phi_e\tau_R$ and thus $M\phi_e\tau_R =_E N\phi_e\tau_R$, that is $M\phi_e\sigma_{\tilde{N_k}}\tau_R =_E N\phi_e\sigma_{\tilde{N_k}}\tau_R$.

Proof of Claim 1: This result is proved by inspection of the rewrite rules, using the fact that the decryption key sk_T is not deducible. More precisely, assume that $M\phi_e\tau \to U$ for some $\tau \in \{\tau_R, \tau_L\}$. It means that there exists a rewriting rule $l \to r \in \mathcal{R}_E$ and a position p such that $M\phi_e\tau|_p =_{E'} l\theta$ for some θ. p cannot occur below M since $\phi_e\tau$ is in normal form. If $M|_p = l\theta'$ for some θ' then we conclude that we can rewrite M as expected. The only interesting case is thus when $M|_p$ is not an instance of l but $M\phi_e\tau|_p$ is. By inspection of the rules, $l \to r$ can only correspond to one of the three equations E5, E6 or E7. The case of Equations E5 or E6 is ruled out by the fact that sk_T is not deducible from $\phi_e\tau$. The last case is when the rule corresponding to Equation E7 is applied. Then it must be the case that $M|_p = x * y$ with x, y variables of $\mathrm{dom}(\phi_e)$. By construction of ϕ_e, we have that $(x * y)\phi_e \to z\phi_e$ (applying the rule corresponding to Equation E7), thus the result.

Proof of Claim 2: Assume by contradiction that there exist M, N two terms such that $M\phi_e\tau =_{E'} N\phi_e\tau$ for some $\tau \in \{\tau_R, \tau_L\}$ and $M\phi_e \neq_{E'} N\phi_e$. Consider M, N two minimal terms that satisfy this property. By case inspection, it must be the case that M and N are both variables. Thus we have $x\phi_e\tau =_{E'} y\phi_e\tau$ and $x\phi_e \neq_{E'} y\phi_e$ with $x, y \in \mathrm{dom}(\phi_e)$, $x \neq y$. The head symbol of $x\phi_e\tau$ must be penc. Then by construction of ϕ_e, τ does not change the randomness used in penc and the randomness uniquely determines the variable, which implies $x = y$, contradiction. □

We now demonstrate that tallying valid ballots yields the same result in both worlds.

Lemma 11. *Let ℓ be a number of candidates. Let $N_3, \ldots, N_k$ be terms, free w.r.t. $sk_T, d, r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}$. Let $\theta_{\tilde{N_k}} = \{{}^{N_k}/_{y_k} \mid k \in \{3, \ldots, n\}\}$ such that $N_i\theta_{\tilde{N_k}}\sigma\tau$ is a valid ballot for any $\tau \in \{\tau_L, \tau_R\}$. Let $\sigma = \{{}^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}, {}^{ballot_1}/_{y_1}, {}^{ballot_2}/_{y_2}\}$. Then*

$$result_i\ \theta_{\tilde{N_k}}\sigma\tau_L =_E result_i\ \theta_{\tilde{N_k}}\sigma\tau_R$$

and both $result_j\ \theta_{\tilde{N_k}}\sigma\tau_L$ and $result_j\ \theta_{\tilde{N_k}}\sigma\tau_R$ are terms built from constants one *and* zero *by application of the function symbol $+$.*

Proof. We first define $N'_i = N_i\theta_{\tilde{N}_k}$. By Lemma 9, we know that $\pi_j(N'_i\sigma\tau_L) =_E \mathsf{penc}(z_{\mathsf{pk}}, U^i_j, V^i_j)\sigma\tau_L$ for some free terms U^i_j, V^i_j. By Lemma 10, we know that $\phi_1\tau_L \approx_s \phi_1\tau_R$ thus we can deduce $\pi_j(N'_i\sigma\tau_R) =_E \mathsf{penc}(z_{\mathsf{pk}}, U^i_j, V^i_j)\sigma\tau_R$. The equational theory ensure that $\mathsf{penc}(K, U, V) =_E \mathsf{penc}(K', U', V')$ implies $K =_E K'$, $U =_E U'$, and $V =_E V'$. Thus we deduce $V^i_j\sigma\tau_L =_E V^i_j\sigma\tau_R$. Therefore, we get that $\mathit{result}_j\ \theta_{\tilde{N}_k}\sigma\tau_L =_E x^{\mathsf{vote}}_{1,j}\tau_L + x^{\mathsf{vote}}_{2,j}\tau_L + +(V^3_j + \cdots + V^k_j)\sigma\tau_L =_E x^{\mathsf{vote}}_{1,j}\tau_R + x^{\mathsf{vote}}_{2,j}\tau_R + +(V^3_j + \cdots + V^k_j)\sigma\tau_R =_E \mathit{result}_j\ \theta_{\tilde{N}_k}\sigma\tau_R$.

Moreover, $V^i_j\sigma\tau \in \{\mathsf{one}, \mathsf{zero}\}$ is ensured by the fact that $N_i\theta_{\tilde{N}_k}\sigma\tau$ is a valid ballot. Therefore we deduce that both $\mathit{result}_j\ \theta_{\tilde{N}_k}\sigma\tau_L$ and $\mathit{result}_j\ \theta_{\tilde{N}_k}\sigma\tau_R$ are terms built from constants one and zero by application of the function symbol $+$. □

We finally show that the encrypted ballots of honest voters and the partial decryptions do not leak any information to the adversary.

Lemma 12. *Let* $\phi_6 = \nu\, sk_T, d, r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}\ .\ (\{^{ballot_1}/_{x_1}\} \mid \{^{ballot_2}/_{x_2}\} \mid \{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\} \mid \{^{\mathsf{partial}_j}/_{x^{\mathsf{partial}}_j} \mid 1 \le j \le \ell\})$. *We have* $\phi_6\sigma_{\tilde{N}_k}\tau_L \approx_s \phi_6\sigma_{\tilde{N}_k}\tau_R$.

The proof is very similar to the proof of Lemma 10

Proof. First, we decompose ϕ_6 and consider $\phi = \nu\,\tilde{n}.\theta$ where $\tilde{n} = \{sk_T, d, r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}\}$ and $\theta = \{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\} \mid \left\{\{^{\mathsf{partial}_j}/_{x_{\mathsf{partial}_j}}\} \mid \{^{ciph_{i,j}}/_{x_{ciph_{i,j}}}\} \mid \{^{spk_{i,j}}/_{x_{spk_{i,j}}}\} \mid \{^{\widehat{spk}_i}/_{x_{\widehat{spk}_i}}\} \;\middle|\; i \in \{1,2\} \wedge 1 \le j \le \ell\right\}$. It follows immediately that $\phi_6\tau_L \approx_s \phi_6\tau_R$ if and only if $\phi\tau_L \approx_s \phi\tau_R$.

Secondly, witness that the adversary can arbitrarily combine ciphertexts from the frame – namely, ciphertexts $ciph_{1,1}, ciph_{2,1}, \ldots, ciph_{1,\ell}, ciph_{2,\ell}$ – with ciphertexts in the frame or freshly constructed ciphertexts, we enrich the frame ϕ with any such combination of ciphertexts. Formally, for any $\alpha_j, \beta_j \in \mathbb{N}$ and terms P, R we define $C_{\alpha_1,\ldots,\alpha_\ell,\beta_1,\ldots,\beta_\ell,\alpha_4,P,R}$ as follows:

$$\mathsf{penc}(\mathsf{pk}(sk_T),\ R \circ \textstyle\bigcirc_{1\le j\le \ell} r^{\alpha_j}_{1,j} \circ r^{\beta_j}_{2,j},\ P + \sum_{1\le j\le \ell} \alpha_j.x^{\mathsf{vote}}_{1,j} + \beta_j.x^{\mathsf{vote}}_{2,j})$$

We define the extended frame ϕ_e below.

$$\begin{aligned}\phi_e = \nu\,\tilde{n}.(\theta \mid \{^{C_{\alpha_1,\alpha_2,\alpha_3,\alpha_4,P,R}}/_{x_{\alpha_1,\alpha_2,\alpha_3,\alpha_4,P,R}} \mid \alpha_1,\alpha_2,\alpha_3,\alpha_4 \in \mathbb{N} \text{ and terms}\\ P, R \text{ s.t. } (\mathrm{fn}(P)\cup\mathrm{fn}(R))\cap\tilde{n} = \emptyset, \mathrm{fv}(P,R) \subseteq \mathrm{dom}(\phi_e) \text{ with no cycle}\})\end{aligned}$$

Note that ϕ_e is infinite. By Lemma 6, it is sufficient to show $\phi_e\tau_L \approx_s \phi_e\tau_R$. Let $\phi'_e = \phi_e\sigma_{\tilde{N}_k}$. We introduce the following two claims.

Claim 3. *Let* M *be a term such that* $\mathrm{fv}(M) \cap (\mathrm{fv}(\phi'_e) \setminus \mathrm{dom}(\phi'_e)) = \emptyset$ *and* $\mathrm{fn}(M) \cap \tilde{n} = \emptyset$. *If* $M\phi'_e\tau \to U$ *for some* $\tau \in \{\tau_R, \tau_L\}$ *then there exists* N *such that* $U =_{E'} N\phi'_e\tau$ *and* $M\phi'_e\tau' \to N\phi'_e\tau'$ *for any* $\tau' \in \{\tau_R, \tau_L\}$.

Claim 4. *Let M, N be two terms such that* $(\mathrm{fv}(M) \cup \mathrm{fv}(N)) \cap (\mathrm{fv}(\phi'_e) \setminus \mathrm{dom}(\phi'_e)) = \emptyset$ *and* $\mathrm{fn}(M,N) \cap \tilde{n} = \emptyset$. *If* $M\phi'_e\tau =_{E'} N\phi'_e\tau$ *for some* $\tau \in \{\tau_R, \tau_L\}$ *then* $M\phi'_e =_{E'} N\phi'_e$.

The above claims allow the construction of our proof. Let M, N be two terms such that $\mathrm{fn}(M,N) \cap \tilde{n} = \emptyset$ and $M\phi'_e\sigma_{\tilde{N_k}}\tau_L =_E N\phi'_e\sigma_{\tilde{N_k}}\tau_L$. We assume (possibly by renaming) that $(\mathrm{fv}(M) \cup \mathrm{fv}(N)) \cap (\mathrm{fv}(\phi'_e) \backslash \mathrm{dom}(\phi'_e)) = \emptyset$. We have $M\phi'_e\tau_L =_E N\phi'_e\tau_L$. Thus $(M\phi'_e\tau_L)\downarrow =_{E'} (N\phi'_e\tau_L)\downarrow$. Applying repeatedly Claim 3, we deduce that there exists M' such that $(M\phi'_e\tau_L)\downarrow = M'\phi'_e\tau_L$ and $M\phi'_e\tau_R \to^* M'\phi'_e\tau_R$. Similarly, there exists N' such that $(N\phi'_e\tau_L)\downarrow = N'\phi'_e\tau_L$ and $N\phi'_e\tau_R \to^* N'\phi'_e\tau_R$. From $M'\phi'_e\tau_L =_{E'} N'\phi'_e\tau_L$ and Claim 4, we deduce $M'\phi'_e =_{E'} N'\phi'_e$. Therefore $M'\phi'_e\tau_R =_{E'} N'\phi'_e\tau_R$ and thus $M\phi'_e\tau_R =_E N\phi'_e\tau_R$, that is $M\phi'_e\sigma_{\tilde{N_k}}\tau_R =_E N\phi'_e\sigma_{\tilde{N_k}}\tau_R$.

Proof of Claim 3: This result is proved by inspection of the rewrite rules, using the fact that the decryption key sk_T is not deducible. More precisely, assume that $M\phi'_e\tau \to U$ for some $\tau \in \{\tau_R, \tau_L\}$. It means that there exists a rewriting rule $l \to r \in \mathcal{R}_E$ and a position p such that $M\phi'_e\tau|_p =_{E'} l\theta$ for some θ. p cannot occur below M since $\phi'_e\tau$ is in normal form. If $M|_p = l\theta'$ for some θ' then we conclude that we can rewrite M as expected. The only interesting case is thus when $M|_p$ is not an instance of l but $M\phi'_e\tau|_p$ is. By inspection of the rules, $l \to r$ can only correspond to one of the three equations E5, E6 or E7. The case of Equations E5 is ruled out by the fact that sk_T is not deducible from $\phi'_e\tau$. For Equation E6, it must be the case that $M\phi'_e|_p = result_j\sigma_{\tilde{N_k}}$. Using Lemma 11, we deduce that $M\phi'_e|_p\tau \to R$ modulo E' where R is a sum of ones and zero. Therefore $M\phi'_e\tau \to M[R]_p\phi'_e\tau$. The last case is when the rule corresponding to Equation E7 is applied. Then it must be the case that $M|_p = x * y$ with x, y variables of $\mathrm{dom}(\phi'_e)$. By construction of ϕ'_e, we have that $(x * y)\phi'_e \to z\phi'_e$ (applying the rule corresponding to Equation E7), thus the result.

Proof of Claim 4: Assume by contradiction that there exist M, N two terms such that $M\phi'_e\tau =_{E'} N\phi'_e\tau$ for some $\tau \in \{\tau_R, \tau_L\}$ and $M\phi'_e \neq_{E'} N\phi'_e$. Consider M, N two minimal terms that satisfy this property. By case inspection, it must be the case that M and N are both variables. Thus we have $x\phi'_e\tau =_{E'} y\phi'_e\tau$ and $x\phi'_e \neq_{E'} y\phi'_e$ with $x, y \in \mathrm{dom}(\phi'_e)$, $x \neq y$. The head symbol of $x\phi'_e\tau$ must be penc or partial. Assume first that the head symbol of $x\phi'_e\tau$ is penc. Then by construction of ϕ'_e, τ does not change the randomness used in penc and the randomness uniquely determines the variable, which implies $x = y$, contradiction. Assume now that the head symbol of $x\phi'_e\tau$ is partial. Then it must be the case that $tally_{j_1}\sigma_{\tilde{N_k}}\tau =_{E'} tally_{j_2}\sigma_{\tilde{N_k}}\tau$ while $tally_{j_1}\sigma_{\tilde{N_k}} \neq_{E'} tally_{j_2}\sigma_{\tilde{N_k}}$. This would require $x_{ciph_{i,j_1}}\tau = x'_{ciph_{i',j_2}}\tau$ for some i, i', which is excluded due to the randomness. □

Figure 6 Partial evolutions of the Helios process specification

We introduce some partial evolutions of the Helios process specification:

$$
\begin{array}{lcl}
A^1 & = & \nu\, sk_T, a_2, d, r_{1,1}, \ldots, r_{1,\ell}, y_1 \,.\, (\_ \mid \{^{ballot_1}/_{y_1}\} \mid \{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\}) \\
A^2 & = & A^1[\_ \mid \{^{ballot_1}/_{x_1}\}] \\
A^3 & = & \nu\, sk_T, d, r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}, y_1, y_2 \,.\, (\_ \mid \\
 & & \quad \{^{ballot_1}/_{y_1}\} \mid \{^{ballot_2}/_{y_2}\} \mid \{^{ballot_1}/_{x_1}\} \mid \{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\}) \\
A^4 & = & A^3[\_ \mid \{^{ballot_2}/_{x_2}\}] \\
A^5 & = & A^4[\nu\, y_{\mathsf{partial}} \,.\, (\_ \mid \{^{partials}/_{y_{\mathsf{partial}}}\}] \\
A^6 & = & A^5[\_ \mid \{^{partials}/_{x_{\mathsf{partial}}}\})] \\
A^7 & = & A^6[\{^{results}/_{x_{\mathsf{result}}}\}]
\end{array}
$$

$$
\begin{array}{lcl}
BB^1_n & = & \overline{c}\langle y_1 \rangle \,.\, BB^2_n \\
BB^2_n & = & a_2(y_2) \,.\, BB^3_n \\
BB^3_n & = & \overline{c}\langle y_2 \rangle \,.\, BB^{'}_{3,n} \\
BB^{'}_{j,n} & = & a_j(y_j) \,.\, \text{if } \phi^{\mathsf{sol}}_{\ell,j-1}\{^{y_j}/_{y_{\mathsf{ballot}}}\} \text{ then} \\
 & & \cdots\ a_n(y_n) \,.\, \text{if } \phi^{\mathsf{sol}}_{\ell,n-1}\{^{y_n}/_{y_{\mathsf{ballot}}}\} \text{ then} \\
 & & BB^4_n \\
BB^{''}_{j,n} & = & \text{if } \phi^{\mathsf{sol}}_{\ell,j-1}\{^{y_j}/_{y_{\mathsf{ballot}}}\} \text{ then} \\
 & & a_{j+1}(y_{j+1}) \,.\, \text{if } \phi^{\mathsf{sol}}_{\ell,j}\{^{y_{j+1}}/_{y_{\mathsf{ballot}}}\} \text{ then} \\
 & & \cdots\ a_n(y_n) \,.\, \text{if } \phi^{\mathsf{sol}}_{\ell,n-1}\{^{y_n}/_{y_{\mathsf{ballot}}}\} \text{ then} \\
 & & BB^4_n \\
BB^4_n & = & \overline{d}\langle (\!| tally_1, \ldots, tally_\ell |\!) \rangle \,.\, BB^5_n \\
BB^5_n & = & d(y_{\mathsf{partial}}) \,.\, BB^6_n \\
BB^6_n & = & \overline{c}\langle y_{\mathsf{partial}} \rangle \,.\, BB^7_n \\
BB^7_n & = & \overline{c}\langle (\!| \mathsf{dec}(\pi_1(y_{\mathsf{partial}}), tally_\ell), \ldots, \mathsf{dec}(\pi_\ell(y_{\mathsf{partial}}), tally_\ell) |\!) \rangle
\end{array}
$$

$$
\begin{array}{lcl}
T^1_\ell & = & \overline{d}\langle partials \rangle
\end{array}
$$

where $partials = (\!| \mathsf{partial}(sk_T, tally_1), \ldots, \mathsf{partial}(sk_T, tally_\ell) |\!)$ and $results = (\!| \mathsf{dec}(\mathsf{partial}(sk_T, tally_1), tally_1), \ldots, \mathsf{dec}(\mathsf{partial}(sk_T, tally_\ell), tally_\ell) |\!)$.

B.4 Proof of Theorem 1

We introduce some partial evolutions of the Helios process specification in Figure 6 and define a relation $\mathcal{R}$ between processes in Figure 7. We clearly have that $A^{\phi}_{\ell,n}[V\{^{a_1}/_{x_{\mathsf{auth}}}\}\sigma \mid V\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma']\ \mathcal{R}\ A^{\phi}_{\ell,n}[V\{^{a_1}/_{x_{\mathsf{auth}}}\}\sigma' \mid V\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma]$. We now wish to show that $\mathcal{R} \cup \mathcal{R}^{-1}$ satisfies the three properties of Definition 2. By symmetry we focus on $\mathcal{R}$. Overwriting the definition, we may say that a term N is a valid ballot if both $N\sigma\tau_L$ and $N\sigma\tau_R$ are valid ballots, where σ is defined Figure 7.

Static equivalence. We must show for all extended processes A and B, where $A\ \mathcal{R}\ B$, that $A \approx_s B$. By Lemma 6, it is sufficient to show $A^7\tau\tau_L \approx_s A^7\tau\tau_R$

Figure 7 Definition of the relation $\mathcal{R}$

Consider the smallest relation $\mathcal{R}$ which is closed under structural equivalence and includes the following pairs of extended processes, where for all $3 \leq j \leq n$, terms M, terms $N_1, \ldots, N_j$, substitutions $\sigma = \{^{N_k}/_{y_k} \mid k \in \{3, \ldots, n\}\}$ and distinct variables $x_{\mathsf{partial}}, x_{\mathsf{result}}, x_1, x_2$ such that $N_j\sigma\tau_L$ and $N_j\sigma\tau_R$ are valid ballot, $\mathrm{fv}(M) \cup \bigcup_{3\leq i\leq j} \mathrm{fv}(N_i) \subseteq \mathrm{dom}(A^4)$ and $(\mathrm{fn}(M) \cup \bigcup_{3\leq i\leq j} \mathrm{fn}(N_i)) \cap \mathrm{bn}(A^4) = \emptyset$.

$$A^{\phi^{\mathsf{sol}}}_{\ell,n}[V\{^{a_1}/_{x_{\mathsf{auth}}}\}\sigma \mid V\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma'], \quad A^{\phi^{\mathsf{sol}}}_{\ell,n}[V\{^{a_1}/_{x_{\mathsf{auth}}}\}\sigma' \mid V\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma] \tag{R1}$$

$$A^1[V\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid BB^1_n \mid T_\ell]\tau_L, \quad A^1[V\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma \mid BB^1_n \mid T_\ell]\tau_R \tag{R2}$$

$$A^2[V\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid BB^2_n \mid T_\ell]\tau_L, \quad A^2[V\{^{a_2}/_{x_{\mathsf{auth}}}\}\sigma \mid BB^2_n \mid T_\ell]\tau_R \tag{R3}$$

$$A^3[BB^3_n \mid T_\ell]\tau_L, \quad A^3[BB^3_n \mid T_\ell]\tau_R \tag{R4}$$

$$A^4[BB^{'}_{j,n}\{^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\} \mid T_\ell]\tau_L, \quad A^4[BB^{'}_{j,n}\{^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\} \mid T_\ell]\tau_R \tag{R5}$$

$$A^4[BB^{''}_{j,n}\{^{N_k}/_{y_k} \mid k \in \{3, \ldots, j-1\}\}\{^{M}/_{y_j}\} \mid T_\ell]\tau_L, \quad A^4[BB^{''}_{j,n}\{^{N_k}/_{y_k} \mid k \in \{3, \ldots, j-1\}\}\{^{M}/_{y_j}\} \mid T_\ell]\tau_R \tag{R6}$$

$$A^4[0 \mid T_\ell]\tau_L, \quad A^4[0 \mid T_\ell]\tau_R \tag{R7}$$

$$A^4[BB^4_n\tau \mid T_\ell]\tau_L, \quad A^4[BB^4_n\tau \mid T_\ell]\tau_R \tag{R8}$$

$$A^4[BB^5_n \mid T^1_\ell]\tau\tau_L, \quad A^4[BB^5_n \mid T^1_\ell]\tau\tau_R \tag{R9}$$

$$A^5[BB^6_n]\tau\tau_L, \quad A^5[BB^6_n]\tau\tau_R \tag{R10}$$

$$A^6[BB^7_n]\tau\tau_L, \quad A^6[BB^7_n]\tau\tau_R \tag{R11}$$

$$A^7\tau\tau_L, \quad A^7\tau\tau_R \tag{R12}$$

for any $N_3, \ldots, N_n$ valid ballots. Let $\phi_7 = \nu\, sk_T, d, r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}\ .\ (\{^{ballot_1}/_{x_1}\} \mid \{^{ballot_2}/_{x_2}\} \mid \{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\} \mid \{^{(\!|partial_1, \ldots, partial_n|\!)}/_{x_{\mathsf{partial}}}\} \mid \{^{(\!|result_1, \ldots, result_n|\!)}/_{x_{\mathsf{result}}}\})$. We have to show $\phi_7\sigma_{N_3,\ldots,N_n}\tau_L \approx_s \phi_7\sigma_{N_3,\ldots,N_n}\tau_R$. By Lemma 11, we deduce that $(\!|result_1, \ldots, result_\ell|\!)\sigma_{N_3,\ldots,N_n}\tau_L = (\!|result_1, \ldots,$

$result_\ell)\sigma_{N_3,\ldots,N_n}\tau_R$ and is equal to a constant (always deducible) term. Thus by Lemma 5, it is sufficient to show that $\phi_6\sigma_{N_3,\ldots,N_n}\tau_L \approx_s \phi_6\sigma_{N_3,\ldots,N_n}\tau_R$, where ϕ_6 as defined in Lemma 12. We conclude by Lemma 12.

Internal reductions. We must show for all extended processes A and B, where $A \mathcal{R} B$, that if $A \to A'$ for some A', then $B \to^* B'$ and $A' \mathcal{R} B'$ for some B'. We observe that if $A \equiv A^1[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid BB^1_n \mid T_\ell]\tau_L$ and $B \equiv A^1[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma \mid BB^1_n \mid T_\ell]\tau_R$ – that is, $A \mathcal{R} B$ by (R2) – then there is no extended process A' such that $A \to A'$; similarly, for (R4), (R5), (R7), (R10), (R11) and (R12). We proceed by case analysis on the remaining cases.

(R1) We have $A \equiv A^{\phi^{\mathsf{sol}}}_{\ell,n}[V\{{}^{a_1}/_{x_{\mathsf{auth}}}\}\sigma \mid V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma']$ and $B \equiv A^{\phi^{\mathsf{sol}}}_{\ell,n}[V\{{}^{a_1}/_{x_{\mathsf{auth}}}\}\sigma' \mid V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma]$. If $A \to A'$, then it must be the case that $A \equiv C[\overline{a_1}\langle y_1\rangle.0 \mid a_1(y_1).BB^1_n]\tau_L$ and $A' \equiv C[0 \mid BB^1_n]\tau_L$, where $C[\_] = A^1[\nu\, a_1.(\_ \mid V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid T_\ell)]$. It follows from $B \equiv C'[\overline{a_1}\langle y_1\rangle.0 \mid a_1(y_1).BB^1_n]\tau_R$, that $B \to B'$, where $C'[\_] = A^1[\nu\, a_1.(\_ \mid V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma \mid T_\ell)]$ and $B' = A^1[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma \mid BB^1_n \mid T_\ell]\tau_R$. Since $A^1[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid BB^1_n \mid T_\ell]\tau_L \mathcal{R} B'$ and $A' \equiv A^1[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid BB^1_n \mid T_\ell]\tau_L$, we derive $A' \mathcal{R} B'$ by the closure of $\mathcal{R}$ under structural equivalence.

(R3) This case is similar to (R1). We have $A \equiv A^2[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid BB^2_n \mid T_\ell]\tau_L$ and $B \equiv A^2[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma \mid BB^2_n \mid T_\ell]\tau_R$. If $A \to A'$, then it must be the case that $A \equiv C[\overline{a_2}\langle y_2\rangle.0 \mid a_2(y_2).BB^3_n]\tau_L$ and $A' \equiv C[0 \mid BB^3_n]\tau_L$, where $C[\_] = A^3[\nu\, a_2.(\_ \mid T_\ell)]$. It follows from $B \equiv C[\overline{a_2}\langle y_2\rangle.0 \mid a_2(y_2).BB^3_n]\tau_R$, that $B \to B'$, where $B' = A^3[BB^3_n \mid T_\ell]\tau_R$. Since $A^3[BB^3_n \mid T_\ell]\tau_L \mathcal{R} B'$ and $A' \equiv A^3[BB^3_n \mid T_\ell]\tau_L$, we derive $A' \mathcal{R} B'$ by the closure of $\mathcal{R}$ under structural equivalence.

(R6) We have $A \equiv A^4[BB''_{j,n}\{{}^{N_k}/_{y_k} \mid k \in \{3,\ldots,j-1\}\}\{{}^{M}/_{y_j}\} \mid T_\ell]\tau_L$ and $B \equiv A^4[BB''_{j,n}\{{}^{N_k}/_{y_k} \mid k \in \{3,\ldots,j-1\}\}\{{}^{M}/_{y_j}\} \mid T_\ell]\tau_R$ for some integer $j \in \{3,\ldots,n\}$, valid ballots $N_3,\ldots,N_{j-1}$ and term M such that $\mathrm{fv}(M) \cup \bigcup_{3\le i\le j-1}\mathrm{fv}(N_i) \subseteq \mathrm{dom}(A^4)$ and $(\mathrm{fn}(M)\cup\bigcup_{3\le i\le j-1}\mathrm{fn}(N_i))\cap\mathrm{bn}(A^4) = \emptyset$.

If $A \to A'$, then it must be the case that $A \equiv C[\text{if } \phi^{\mathsf{sol}}_{\ell,j-1}\{{}^{M}/_{y_{\mathsf{ballot}}}, {}^{ballot_1}/_{y_1}, {}^{ballot_2}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\} \text{ then } P \text{ else } 0]\tau_L$, where $C[\_] = A^4[\_ \mid T_\ell]$. Furthermore, if $j < n$, then $P = BB'_{j+1,n}\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3,\ldots,j-1\}\}\{{}^{M}/_{y_j}\}$; otherwise $P = BB^4_n\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3,\ldots,j-1\}\}\{{}^{M}/_{y_j}\}$. We also have $B \equiv C[\text{if } \phi^{\mathsf{sol}}_{\ell,j-1}\{{}^{M}/_{y_{\mathsf{ballot}}}, {}^{ballot_1}/_{y_1}, {}^{ballot_2}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\} \text{ then } P \text{ else } 0]\tau_R$.

Let $\sigma_R = \{{}^{ballot_1\tau_R}/_{x_1}, {}^{ballot_2\tau_R}/_{x_2}, {}^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\}$, we have $ballot_1\tau_R$ is syntactically equal to $x_1\sigma_R$ and $ballot_2\tau_R$ is syntactically equal to $x_2\sigma_R$, it follows that $B \equiv C[\text{if } \phi^{\mathsf{sol}}_{\ell,j-1}\{{}^{M}/_{y_{\mathsf{ballot}}}, {}^{x_1\sigma_R}/_{y_1}, {}^{x_2\sigma_R}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\} \text{ then } P \text{ else } 0]\tau_R$ and, moreover, since $\varphi(C[0]\tau_R) = \nu\, sk_T, d, r_{1,1},\ldots,r_{1,\ell}, r_{2,1},\ldots,r_{2,\ell}, y_1, y_2.\sigma_R$ we have $B \equiv C[\text{if } \phi^{\mathsf{sol}}_{\ell,j-1}\{{}^{M}/_{y_{\mathsf{ballot}}}, {}^{x_1}/_{y_1}, {}^{x_2}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\} \text{ then } P \text{ else } 0]\tau_R$. We proceed by case analysis on the structure of A':

– If $A' \equiv C[P]\tau_L$, then by closure of internal reduction under structural equivalence we have $C[\text{if } \phi^{\mathsf{sol}}_{\ell,j-1}\{^{M}/_{y_{\mathsf{ballot}}}, {}^{x_1}/_{y_1}, {}^{x_2}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\} \text{ then } P \text{ else } 0]\tau_L \rightarrow C[P]\tau_L$ because $ballot_1\tau_L$ is syntactically equal to $x_1\sigma_L$, $ballot_2\tau_L$ is syntactically equal to $x_2\sigma_L$ and $\varphi(C[0]\tau_L) = \nu\, sk_T, d, r_{1,1}, \ldots, r_{1,\ell}, r_{2,1}, \ldots, r_{2,\ell}, y_1, y_2.\sigma_L$, where $\sigma_L = \{{}^{ballot_1\tau_L}/_{x_1}, {}^{ballot_2\tau_L}/_{x_2}, {}^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}\}$.

  Assume A and B satisfy the preconditions of Corollary 1, it follows that $B \rightarrow B' = A^4[P \mid T_\ell]\tau_R$. We now prove our assumption. Since $A \mathcal{R} B$, it follows by Condition 1 of Definition 2 that $A \approx_s B$. Let $\phi = \phi^{\mathsf{sol}}_{\ell,j-1}\{^{M}/_{y_{\mathsf{ballot}}}, {}^{x_1}/_{y_1}, {}^{x_2}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\}$. By inspection of $\phi^{\mathsf{sol}}_{\ell,j-1}$, we have $\mathrm{fn}(\phi) = \mathrm{fn}(M) \cup \bigcup_{3 \leq i \leq j-1} \mathrm{fn}(N_i)$ and since $\mathrm{bn}(C) = \mathrm{bn}(A^4)$ it follows that $\mathrm{bn}(C) \cap \mathrm{fn}(\phi) = \emptyset$; we also have $\mathrm{fv}(\phi) = \{x_1, x_2, z_{\mathsf{pk}}\} \cup \mathrm{fv}(M) \cup \bigcup_{3 \leq i \leq j-1} \mathrm{fv}(N_i)$ and since $\mathrm{dom}(C) = \{x_1, x_2, z_{\mathsf{pk}}\}$ it follows that $\mathrm{fv}(\phi) \subset \mathrm{dom}(C)$. We have shown that the preconditions of Corollary 1 are satisfied, hence $B \rightarrow B' = A^4[P \mid T_\ell]\tau_R$. It remains to show $A' \mathcal{R} B'$.

  We know $[\![\phi^{\mathsf{sol}}_{\ell,j-1}\{^{M}/_{y_{\mathsf{ballot}}}, {}^{x_1}/_{y_1}, {}^{x_2}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\}\sigma_L]\!] = \mathsf{true}$ and $[\![\phi^{\mathsf{sol}}_{\ell,j-1}\{^{M}/_{y_{\mathsf{ballot}}}, {}^{x_1}/_{y_1}, {}^{x_2}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\}\sigma_R]\!] = \mathsf{true}$; it follows, for $\tau \in \{\tau_L, \tau_R\}$, that $[\![\phi^{\mathsf{sol}}_{\ell,j-1}\{^{M}/_{y_{\mathsf{ballot}}}\}\{^{\mathsf{pk}(sk_T)}/_{z_{\mathsf{pk}}}, {}^{ballot_1}/_{y_1}, {}^{ballot_2}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\}\tau]\!] = \mathsf{true}$ and we know that M is a valid ballot. We continue by case analysis on the structure of P:

  1. If $P = BB^{'}_{j+1,n}\{^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\}\{^{M}/_{y_j}\}$, then we have $j < n$. Let $j' = j + 1$ and $N_j = M$, observe $P = BB^{'}_{j',n}\{^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j\}\}$ and $A^4[BB^{'}_{j',n}\{^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j\}\} \mid T_\ell]\tau_L \mathcal{R} B'$. The result $A' \mathcal{R} B'$ follows by closure of $\mathcal{R}$ under structural equivalence.
  2. If $P = BB^4_n\{^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\}\{^{M}/_{y_j}\}$, then it must be the case that $j = n$. Let $N_j = M$ and hence $P = BB^4_n\tau$. Since $A^4[BB^4_n\tau \mid T_\ell]\tau_L \mathcal{R} B'$ and $A' \equiv A^4[BB^4_n\tau \mid T_\ell]\tau_L$, we derive $A' \mathcal{R} B'$ by the closure of $\mathcal{R}$ under structural equivalence.

– $A' \equiv C[0]\tau_L$, then similarly to above we have $C[\text{if } \phi^{\mathsf{sol}}_{\ell,j-1}\{^{M}/_{y_{\mathsf{ballot}}}, {}^{x_1}/_{y_1}, {}^{x_2}/_{y_2}, {}^{N_3}/_{y_3}, \ldots, {}^{N_{j-1}}/_{y_{j-1}}\} \text{ then } P \text{ else } 0]\tau_L \rightarrow C[0]\tau_L$ and it follows by Corollary 1 that $B \rightarrow B' = C[0]\tau_R$. Since $A^4[0 \mid T_\ell] \mathcal{R} B'$ and $A' \equiv A^4[0 \mid T_\ell]$, we derive $A' \mathcal{R} B'$ by the closure of $\mathcal{R}$ under structural equivalence.

(R8) We have $A \equiv A^4[BB^4_n\tau \mid T_\ell]\tau_L$ and $B \equiv A^4[BB^4_n\tau \mid T_\ell]\tau_R$, where $BB^4_n = \overline{d}\langle\!(tally_1, \ldots, tally_\ell)\!\rangle \,.\, BB^5_n$ and $T_\ell = d(y_{\mathsf{tally}}) \,.\, \overline{d}\langle\!(\mathsf{partial}(sk_T, \pi_1(y_{\mathsf{tally}})), \ldots, \mathsf{partial}(sk_T, \pi_\ell(y_{\mathsf{tally}})))\!\rangle$. If $A \rightarrow A'$, then it must be the case that $A' \equiv A^4[BB^5_n \mid T^1_\ell]\tau\tau_L$. It follows immediately that $B \rightarrow B'$, where $B' = A^4[BB^5_n \mid T^1_\ell]\tau\tau_R$. We derive $A' \mathcal{R} B'$ by the closure of $\mathcal{R}$ under structural equivalence.

(R9) We have $A \equiv A^4[BB^5_n \mid T^1_\ell]\tau\tau_L$ and $B \equiv A^4[BB^5_n \mid T^1_\ell]\tau\tau_R$. If $A \rightarrow A'$,

then it must be the case that $A \equiv A^5[\overline{d}\langle y_{\mathsf{partial}}\rangle.0 \mid d(y_{\mathsf{partial}}).BB_n^6\}]\tau\tau_L$ and $A' \equiv A^5[0 \mid BB_n^6]\tau\tau_L$. It follows from $B \equiv A^5[d(y_{\mathsf{partial}}).BB_n^6 \mid \overline{d}\langle y_{\mathsf{partial}}\rangle.0]\tau\tau_R$ that $B \to B'$, where $B' = A^5[BB_n^6]\tau\tau_R$. We derive $A' \mathcal{R} B'$ by the closure of $\mathcal{R}$ under structural equivalence.

Labelled reductions. We must show for all extended processes A and B, where $A \mathcal{R} B$, that if $A \xrightarrow{\alpha} A'$ such that $\mathrm{fv}(\alpha) \subseteq \mathrm{dom}(A)$ and $\mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset$, then $B \to^* \xrightarrow{\alpha} \to^* B'$ and $A' \mathcal{R} B'$ for some B'. We observe cases (R1), (R3), (R6), (R7), (R8), (R9) and (R12) cannot be reduced by labelled reductions and proceed by case analysis on the remaining cases.

(R2) We have $A \equiv A^1[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid BB_n^1 \mid T_\ell]\tau_L$ and $B \equiv A^1[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma \mid BB_n^1 \mid T_\ell]\tau_R$. If $A \xrightarrow{\alpha} A'$ such that $\mathrm{fv}(\alpha) \subseteq \mathrm{dom}(A)$ and $\mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset$, then it must be the case that $A \equiv A^1[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid \overline{c}\langle y_1\rangle.BB_n^2 \mid T_\ell]\tau_L$ and $A' \equiv A^2[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid BB_n^2 \mid T_\ell]\tau_L$ for some variable x_1 where $\alpha = \nu\, x_1.\overline{c}\langle x_1\rangle$ and $x_1 \neq z_{\mathsf{pk}}$. It follows from $B \equiv A^1[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma \mid \overline{c}\langle y_1\rangle.BB_n^2 \mid T_\ell]\tau_R$, that $B \xrightarrow{\alpha} B'$ where $B' = A^2[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma \mid BB_n^2 \mid T_\ell]\tau_R$. We have $A^2[V\{{}^{a_2}/_{x_{\mathsf{auth}}}\}\sigma' \mid BB_n^2 \mid T_\ell]\tau_L \mathcal{R} B'$ and by closure of $\mathcal{R}$ under structural equivalence $A' \mathcal{R} B'$.

(R4) We have $A \equiv A^3[BB_n^3 \mid T_\ell]\tau_L$ and $B \equiv A^3[BB_n^3 \mid T_\ell]\tau_R$. If $A \xrightarrow{\alpha} A'$ such that $\mathrm{fv}(\alpha) \subseteq \mathrm{dom}(A)$ and $\mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset$, then it must be the case that $A \equiv A^3[\overline{c}\langle y_2\rangle.BB_{3,n}^{'} \mid T_\ell]\tau_L$ and $A' \equiv A^4[BB_{3,n}^{'} \mid T_\ell]\tau_L$ for some variable x_2, where $\alpha = \nu\, x_2.\overline{c}\langle x_2\rangle$ and $x_2 \notin \{x_1, z_{\mathsf{pk}}\}$. It follows from $B \equiv A^3[\overline{c}\langle y_2\rangle.BB_{3,n}^{'} \mid T_\ell]\tau_R$, that $B \xrightarrow{\alpha} B'$, where $B' = A^4[BB_{3,n}^{'} \mid T_\ell]\tau_R$. Since $A^4[BB_{3,n}^{'} \mid T_\ell]\tau_L = A^4[BB_{j,n}^{'}\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\} \mid T_\ell]\tau_L$ and $A^4[BB_{3,n}^{'} \mid T_\ell]\tau_R = A^4[BB_{j,n}^{'}\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\} \mid T_\ell]\tau_R$ when $j = 3$, we have $A^4[BB_{3,n}^{'} \mid T_\ell]\tau_L \mathcal{R} B'$ and derive $A' \mathcal{R} B'$ by closure of $\mathcal{R}$ under structural equivalence $A' \mathcal{R} B'$.

(R5) We have $A \equiv A^4[BB_{j,n}^{'}\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\} \mid T_\ell]\tau_L$ and $B \equiv A^4[BB_{j,n}^{'}\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\} \mid T_\ell]\tau_R$ for some integer $j \in \{3, \ldots, n\}$ and terms $N_3, \ldots, N_{j-1}$, where $\bigcup_{3 \leq i \leq j-1} \mathrm{fv}(N_i) \subseteq \mathrm{dom}(A^4)$ and $\mathrm{bn}(A^4) \cap \bigcup_{3 \leq i \leq j-1} \mathrm{fn}(N_i) = \emptyset$. If $A \xrightarrow{\alpha} A'$ such that $\mathrm{fv}(\alpha) \subseteq \mathrm{dom}(A)$ and $\mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset$, then it must be the case that $A \equiv A^4[a_j(y_j).BB_{j,n}^{''}\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\} \mid T_\ell]\tau_L$ and $A' \equiv A^4[BB_{j,n}^{''}\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\}\{{}^{M}/_{y_j}\} \mid T_\ell]\tau_L$, where $\alpha = c(M)$ for some term M. It follows from $B \equiv A^4[a_j(y_j).BB_{j,n}^{''}\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\} \mid T_\ell]\tau_R$, that $B \xrightarrow{\alpha} B'$, where $B' = A^4[BB_{j,n}^{''}\{{}^{N_k}/_{y_k} \mid j > 3 \wedge k \in \{3, \ldots, j-1\}\}\{{}^{M}/_{y_j}\} \mid T_\ell]\tau_R$. We have $A^4[BB_{j,n}^{''}\{{}^{N_k}/_{y_k} \mid k \in \{3, \ldots, j-1\}\}\{{}^{M}/_{y_j}\} \mid T_\ell]\tau_L \mathcal{R} B'$, and derive $A' \mathcal{R} B'$ by closure of $\mathcal{R}$ under structural equivalence.

(R10) We have $A \equiv A^5[BB_n^6]\tau\tau_L$ and $B \equiv A^5[BB_n^6]\tau\tau_R$. If $A \xrightarrow{\alpha} A'$ such that $\mathrm{fv}(\alpha) \subseteq \mathrm{dom}(A)$ and $\mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset$, then it must be the case that

$A' \equiv A^6[BB_n^7]\tau\tau_L$ for some variable x_{partial}, where $\alpha = \nu\, x_{\mathsf{partial}}.\overline{c}\langle x_{\mathsf{partial}}\rangle$ and $x_{\mathsf{partial}} \notin \{x_1, x_2, z_{\mathsf{pk}}\}$. It follows immediately that $B \xrightarrow{\alpha} B'$, where $B' = A^6[BB_n^7]\tau\tau_R$. We have $A^6[BB_n^7]\tau\tau_L \;\mathcal{R}\; B'$ and by closure of $\mathcal{R}$ under structural equivalence $A' \;\mathcal{R}\; B'$.

(R11) This case is similar to (R10). We have $A \equiv A^6[BB_n^7]\tau\tau_L$ and $B \equiv A^6[BB_n^7]\tau\tau_R$. If $A \xrightarrow{\alpha} A'$ such that $\mathrm{fv}(\alpha) \subseteq \mathrm{dom}(A)$ and $\mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset$, then it must be the case that $A' \equiv A^7\tau\tau_L$ for some variable x_{result}, where $\alpha = \nu\, x_{\mathsf{result}}.\overline{c}\langle x_{\mathsf{result}}\rangle$ and $x_{\mathsf{result}} \notin \{x_1, x_2, x_{\mathsf{partial}}, z_{\mathsf{pk}}\}$. It follows immediately that $B \xrightarrow{\alpha} B'$, where $B' = A^7\tau\tau_R$. We have $A^7\tau\tau_L \;\mathcal{R}\; B'$ and by closure of $\mathcal{R}$ under structural equivalence $A' \;\mathcal{R}\; B'$.

References

[Adi06] Ben Adida. *Advances in Cryptographic Voting Systems*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2006.

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

[Adi10] Ben Adida. Attacks and Defenses. Helios documentation, `http://documentation.heliosvoting.org/attacks-and-defenses`, 2010.

[AF01] Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *POPL'01: 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 104–115. ACM Press, 2001.

[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.

[AP10] Ben Adida and Olivier Pereira. Private email communication, November 2010.

[AR00] Martín Abadi and Phillip Rogaway. Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption). In *IFIP TCS'00: 1st International Conference on Theoretical Computer Science*, volume 1872 of *LNCS*, pages 3–22. Springer, 2000.

[AR02] Martín Abadi and Phillip Rogaway. Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption). *Journal of Cryptology*, 15(2):103–127, 2002.

[BAF08] Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, 75(1):3–51, February–March 2008.

[BCP+11] David Bernhard, Véronique Cortier, Olivier Pereira, Ben Smyth, and Bogdan Warinschi. Adapting Helios for provable ballot privacy. In *ESORICS'11: 16th European Symposium on Research in Computer Security*, volume 6879 of *LNCS*, pages 335–354. Springer, 2011.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In *CRYPTO'98: 18th International Cryptology Conference*, volume 1462 of *LNCS*, pages 26–45. Springer, 1998.

[Ben96] Josh Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Department of Computer Science, Yale University, 1996.

[Ben06] Josh Benaloh. Simple Verifiable Elections. In *EVT'06: Electronic Voting Technology Workshop*. USENIX Association, 2006.

[Ben07] Josh Benaloh. Ballot Casting Assurance via Voter-Initiated Poll Station Auditing. In *EVT'07: Electronic Voting Technology Workshop*. USENIX Association, 2007.

[Ber12] David Bernhard. Private email communication, 10th March 2012.

[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.

[BHM08] Michael Backes, Cătălin Hriţcu, and Matteo Maffei. Automated Verification of Remote Electronic Voting Protocols in the Applied Pi-calculus. In *CSF'08: 21st Computer Security Foundations Symposium*, pages 195–209. IEEE Computer Society, 2008.

[BPW12] David Bernhard, Olivier Pereira, and Bogdan Warinschi. On Necessary and Sufficient Conditions for Private Ballot Submission. Cryptology ePrint Archive, Report 2012/236, 2012.

[BVQ10] Josh Benaloh, Serge Vaudenay, and Jean-Jacques Quisquater. Final Report of IACR Electronic Voting Committee. International Association for Cryptologic Research. `http://www.iacr.org/elections/eVoting/finalReportHelios_2010-09-27.html`, Sept 2010.

[BY86] Josh Benaloh and Moti Yung. Distributing the Power of a Government to Enhance the Privacy of Voters. In *PODC'86: 5th Principles of Distributed Computing Symposium*, pages 52–62. ACM Press, 1986.

[CCM07] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a Secure Voting System. Technical Report 2007-2081, Cornell University, May 2007. Revised March 2008.

[CCM08] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a Secure Voting System. In *S&P'08: 29th Security and Privacy Symposium*, pages 354–368. IEEE Computer Society, 2008.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *CRYPTO'94: 14th International Cryptology Conference*, volume 839 of *LNCS*, pages 174–187. Springer, 1994.

[CEG88] David Chaum, Jan-Hendrik Evertse, and Jeroen van de Graaf. An Improved Protocol for Demonstrating Possession of Discrete Logarithms and Some Generalizations. In *EUROCRYPT'87: 4th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 304 of *LNCS*, pages 127–141. Springer, 1988.

[CEGP87] David Chaum, Jan-Hendrik Evertse, Jeroen van de Graaf, and René Peralta. Demonstrating Possession of a Discrete Logarithm Without Revealing It. In *CRYPTO'86: 6th International Cryptology Conference*, volume 263 of *LNCS*, pages 200–212. Springer, 1987.

[CFSY96] Ronald Cramer, Matthew K. Franklin, Berry Schoenmakers, and Moti Yung. Multi-Autority Secret-Ballot Elections with Linear Work. In *EUROCRYPT'96: 15th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 1070 of *LNCS*, pages 72–83. Springer, 1996.

[CGMA85] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults. In *FOCS'85: 26th Foundations of Computer Science Symposium*, pages 383–395. IEEE Computer Society, 1985.

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In *EUROCRYPT'97: 16th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 1233 of *LNCS*, pages 103–118. Springer, 1997.

[CP93] David Chaum and Torben P. Pedersen. Wallet Databases with Observers. In *CRYPTO'92: 12th International Cryptology Conference*, volume 740 of *LNCS*, pages 89–105. Springer, 1993.

[CR87] Benny Chor and Michael O. Rabin. Achieving Independence in Logarithmic Number of Rounds. In *PODC'87: 6th Principles of Distributed Computing Symposium*, pages 260–268. ACM Press, 1987.

[CRS05] David Chaum, Peter Y. A. Ryan, and Steve Schneider. A Practical Voter-Verifiable Election Scheme. In *ESORICS'05: 10th European Symposium On Research In Computer Security*, volume 3679 of *LNCS*, pages 118–139. Springer, 2005.

[CS11] Véronique Cortier and Ben Smyth. Attacking and fixing Helios: An analysis of ballot secrecy. In *CSF'11: 24th Computer Security Foundations Symposium*, pages 297–311. IEEE Computer Society, 2011.

[Dag07] Participants of the Dagstuhl Conference on Frontiers of E-Voting. *Dagstuhl Accord*, 2007. `http://www.dagstuhlaccord.org/`.

[DC12a] Yvo Desmedt and Pyrros Chaidos. Applying Divertibility to Blind Ballot Copying in the Helios Internet Voting System. In *ESORICS'12: 17th European Symposium on Research in Computer Security*, volume 7459 of *LNCS*, pages 433–450. Springer, 2012.

[DC12b] Yvo Desmedt and Pyrros Chaidos. Private communication, March 2012.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-Malleable Cryptography. In *STOC'91: 23rd Theory of computing Symposium*, pages 542–552. ACM Press, 1991.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable Cryptography. *Journal on Computing*, 30(2):391–437, 2000.

[DJ01] Ivan Damgård and Mads Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In *PKC'01: 4th International Workshop on Practice and Theory in Public Key Cryptography*, volume 1992 of *LNCS*, pages 119–136. Springer, 2001.

[DJN10] Ivan Damgård, Mads Jurik, and Jesper Buus Nielsen. A Generalization of Paillier's Public-Key System with Applications to Electronic Voting. *International Journal of Information Security*, 9(6):371–385, 2010.

[DKR06] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Coercion-Resistance and Receipt-Freeness in Electronic Voting. In *CSFW'06: 19th Computer Security Foundations Workshop*, pages 28–42. IEEE Computer Society, 2006.

[DKR09] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, July 2009.

[DKR10] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Verifying Privacy-Type Properties of Electronic Voting Protocols: A Taster. In David Chaum, Markus Jakobsson, Ronald L. Rivest, and Peter Y. A. Ryan, editors, *Towards Trustworthy Elections: New Directions in Electronic Voting*, volume 6000 of *LNCS*, pages 289–309. Springer, 2010.

[DLL11] Jannik Dreier, Pascal Lafourcade, and Yassine Lakhnech. Vote-Independence: A Powerful Privacy Notion for Voting Protocols. In *FPS'11: 4th Workshop on Foundations & Practice of Security*, volume 6888 of *LNCS*, pages 164–180. Springer, 2011.

[DRS08] Stéphanie Delaune, Mark D. Ryan, and Ben Smyth. Automatic verification of privacy properties in the applied pi-calculus. In *IFIPTM'08: 2nd Joint iTrust and PST Conferences on Privacy, Trust Management and Security*, volume 263 of *International Federation for Information Processing (IFIP)*, pages 263–278. Springer, 2008.

[ED10] Saghar Estehghari and Yvo Desmedt. Exploiting the Client Vulnerabilities in Internet E-voting Systems: Hacking Helios 2.0 as an Example. In *EVT/WOTE'10: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. USENIX Association, 2010.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[Est07] Est Républicain. June, 18th 2007. Meurthe-et-Moselle edition (Daily French Newspaper).

[FOO92] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A Practical Secret Voting Scheme for Large Scale Elections. In *AUSCRYPT'92: Workshop on the Theory and Application of Cryptographic Techniques*, volume 718 of *LNCS*, pages 244–251. Springer, 1992.

[Fr] *Article L65 of the French electoral code*. `http://www.legifrance.gouv.fr/`.

[Fr10] *Résultat par bureau du premier tour des élections régionales*, 2010. `http://www.monaulnay.com/wp-content/uploads/2010/03/resultat_regionale_par_bureau.pdf`.

[Gen95] Rosario Gennaro. Achieving independence efficiently and securely. In *PODC'95: 14th Principles of Distributed Computing Symposium*, pages 130–136. ACM Press, 1995.

[Gen00] Rosario Gennaro. A Protocol to Achieve Independence in Constant Rounds. *IEEE Transactions on Parallel and Distributed Systems*, 11(7):636–647, 2000.

[HBH10] Stuart Haber, Josh Benaloh, and Shai Halevi. The Helios e-Voting Demo for the IACR. International Association for Cryptologic Research. `http://www.iacr.org/elections/eVoting/heliosDemo.pdf`, May 2010.

[JCJ02] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. Cryptology ePrint Archive, Report 2002/165, 2002.

[JCJ05] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. In *WPES'05: 4th Workshop on Privacy in the Electronic Society*, pages 61–70. ACM Press, 2005. See also `http://www.rsa.com/rsalabs/node.asp?id=2860`.

[JCJ10] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. In David Chaum, Markus Jakobsson, Ronald L. Rivest, and Peter Y. A. Ryan, editors, *Towards Trustworthy Elections: New Directions in Electronic Voting*, volume 6000 of *LNCS*, pages 37–63. Springer, 2010.

[KR05] Steve Kremer and Mark D. Ryan. Analysis of an Electronic Voting Protocol in the Applied Pi Calculus. In *ESOP'05: 14th European Symposium on Programming*, volume 3444 of *LNCS*, pages 186–200. Springer, 2005.

[KRS10] Steve Kremer, Mark D. Ryan, and Ben Smyth. Election verifiability in electronic voting protocols. In *ESORICS'10: 15th European Symposium on Research in Computer Security*, volume 6345 of *LNCS*, pages 389–404. Springer, 2010.

[KT09] Ralf Küsters and Tomasz Truderung. An Epistemic Approach to Coercion-Resistance for Electronic Voting Protocols. In *S&P'09: 30th IEEE Symposium on Security and Privacy*, pages 251–266. IEEE Computer Society, 2009.

[Lan10] Lucie Langer. *Privacy and Verifiability in Electronic Voting*. PhD thesis, Fachbereich Informatik, Technischen Universität Darmstadt, 2010.

[LBD+04] Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang, and Seungjae Yoo. Providing Receipt-Freeness in Mixnet-Based Voting Protocols. In *ICISC'03: 6th International Conference on Information Security and Cryptology*, volume 2971 of *LNCS*, pages 245–258. Springer, 2004.

[LCM08] Adrian Leung, Liqun Chen, and Chris J. Mitchell. On a Possible Privacy Flaw in Direct Anonymous Attestation (DAA). In *Trust'08: 1st International Conference on Trusted Computing and Trust in Information Technologies*, number 4968 in LNCS, pages 179–190. Springer, 2008.

[Liu11] Jia Liu. A Proof of Coincidence of Labeled Bisimilarity and Observational Equivalence in Applied Pi Calculus. `http://lcs.ios.ac.cn/~jliu/papers/LiuJia0608.pdf`, 2011.

[LL90] Arjen K. Lenstra and Hendrik W. Lenstra Jr. Algorithms in Number Theory. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, chapter 12, pages 673–716. MIT Press, 1990.

[LSB+10] Lucie Langer, Axel Schmidt, Johannes Buchmann, Melanie Volkamer, and Alexander Stolfik. Towards a Framework on the Security Requirements for Electronic Voting Protocols. In *Re-Vote'09: First International Workshop on Requirements Engineering for E-Voting Systems*, pages 61–68. IEEE Computer Society, 2010.

[LSBV10] Lucie Langer, Axel Schmidt, Johannes Buchmann, and Melanie Volkamer. A Taxonomy Refining the Security Requirements for Electronic Voting: Analyzing Helios as a Proof of Concept. In *ARES'10: 5th International Conference on Availability, Reliability and Security*, pages 475–480. IEEE Computer Society, 2010.

[NY90] Moni Naor and Moti Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *STOC'90: 22nd Theory of computing Symposium*, pages 427–437. ACM Press, 1990.

[Oka98] Tatsuaki Okamoto. Receipt-Free Electronic Voting Schemes for Large Scale Elections. In *SP'97: 5th International Workshop on Security Protocols*, volume 1361 of *LNCS*, pages 25–35. Springer, 1998.

[Pai10] Miriam Paiola. Extending ProVerif's Resolution Algorithm for Verifying Group Protocols. Master's thesis, Faculty of Mathematical, Physical and Natural Science, University of Padova, 2010.

[PAM10] Olivier Pereira, Ben Adida, and Olivier de Marneffe. Bringing open audit elections into practice: Real world uses of helios. Swiss e-voting workshop, `https://www.e-voting-cc.ch/`

images/sevot10/slides/helios_20100906.pdf. See also http://www.uclouvain.be/crypto/electionmonitor/, 2010.

[PB11] Miriam Paiola and Bruno Blanchet. Automatic Verification of Group Protocols with Unbounded Numbers of Participants and Sessions. Unpublished draft, 2011.

[PB12] Miriam Paiola and Bruno Blanchet. Verification of Security Protocols with Lists: from Length One to Unbounded Length. In *POST'12: First Conference on Principles of Security and Trust*, volume 7215 of *LNCS*, pages 69–88. Springer, 2012.

[Ped91] Torben P. Pedersen. A Threshold Cryptosystem without a Trusted Party. In *EUROCRYPT'91: 10th International Conference on the Theory and Applications of Cryptographic Techniques*, number 547 in LNCS, pages 522–526. Springer, 1991.

[Pfi94] Birgit Pfitzmann. Breaking Efficient Anonymous Channel. In *EUROCRYPT'94: 11th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 950 of *LNCS*, pages 332–340. Springer, 1994.

[PP89] Birgit Pfitzmann and Andreas Pfitzmann. How to Break the Direct RSA-Implementation of Mixes. In *EUROCRYPT'89: 6th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 434 of *lncs*, pages 373–381. sp, 1989.

[Pri10] Princeton University. *Princeton Election Server*, 2010. https://princeton-helios.appspot.com/.

[RS98] Peter Y. A. Ryan and Steve A. Schneider. An Attack on a Recursive Authentication Protocol. A Cautionary Tale. *Information Processing Letters*, 65(1):7–10, 1998.

[RS11] Mark D. Ryan and Ben Smyth. Applied pi calculus. In Véronique Cortier and Steve Kremer, editors, *Formal Models and Techniques for Analyzing Security Protocols*, chapter 6. IOS Press, 2011.

[Rud07] Carsten Rudolph. Covert Identity Information in Direct Anonymous Attestation (DAA). In *SEC'07: 22nd International Information Security Conference*, volume 232 of *International Federation for Information Processing (IFIP)*, pages 443–448. Springer, 2007.

[SC10] Ben Smyth and Véronique Cortier. Attacking ballot secrecy in Helios. YouTube video, linked from http://www.bensmyth.com/publications/10-attacking-helios/, 2010.

[SC11] Ben Smyth and Véronique Cortier. A note on replay attacks that violate privacy in electronic voting schemes. Technical Report RR-7643, INRIA, June 2011. http://hal.inria.fr/inria-00599182/.

[Sch90] Claus-Peter Schnorr. Efficient Identification and Signatures for Smart Cards. In *CRYPTO'89: 9th International Cryptology Conference*, volume 435 of *LNCS*, pages 239–252. Springer, 1990.

[Sch99] Berry Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In *CRYPTO'99: 19th International Cryptology Conference*, volume 1666 of *LNCS*, pages 148–164. Springer, 1999.

[Sch09] Berry Schoenmakers. Voting Schemes. In Mikhail J. Atallah and Marina Blanton, editors, *Algorithms and Theory of Computation Handbook, Second Edition, Volume 2: Special Topics and Techniques*, chapter 15. CRC Press, 2009.

[SG98] Victor Shoup and Rosario Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *EUROCRYPT'97: 17th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 1403 of *LNCS*, pages 1–16. Springer, 1998.

[SG02] Victor Shoup and Rosario Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. *Journal of Cryptology*, 15(2):75–96, 2002.

[Sha71] Daniel Shanks. Class number, a theory of factorization and genera. In *Number Theory Institute*, volume 20 of *Symposia in Pure Mathematics*, pages 415–440. American Mathematical Society, 1971.

[SJ00] Claus-Peter Schnorr and Markus Jakobsson. Security of Signed ElGamal Encryption. In *ASIACRYPT'00: 6th International Conference on the Theory and Application of Cryptology and Information Security*, volume 1976 of *LNCS*, pages 73–89. Springer, 2000.

[SK94] Kazue Sako and Joe Kilian. Secure Voting Using Partially Compatible Homomorphisms. In *CRYPTO'94: 14th International Cryptology Conference*, volume 839 of *LNCS*, pages 411–424. Springer, 1994.

[Smy11] Ben Smyth. *Formal verification of cryptographic protocols with automated reasoning*. PhD thesis, School of Computer Science, University of Birmingham, 2011.

[Smy12] Ben Smyth. Replay attacks that violate ballot secrecy in helios. Cryptology ePrint Archive, Report 2012/185, 2012.

[SRKK10] Ben Smyth, Mark D. Ryan, Steve Kremer, and Mounira Kourjieh. Towards automatic analysis of election verifiability properties. In *ARSPA-WITS'10: Joint Workshop on Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security*, volume 6186 of *LNCS*, pages 165–182. Springer, 2010.

[TY98] Yiannis Tsiounis and Moti Yung. On the Security of ElGamal Based Encryption. In *PKC'98: First International Workshop on Practice and Theory in Public Key Cryptography*, volume 1431 of *LNCS*, pages 117–134. Springer, 1998.

[VG10] Melanie Volkamer and Rüdiger Grimm. Determine the Resilience of Evaluated Internet Voting Systems. In *Re-Vote'09: First International Workshop on Requirements Engineering for E-Voting Systems*, pages 47–54. IEEE Computer Society, 2010.

[Vol09] Melanie Volkamer. *Evaluation of Electronic Voting: Requirements and Evaluation Procedures to Support Responsible Election Authorities*, volume 30 of *Lecture Notes in Business Information Processing*. Springer, 2009.

[War03] Bogdan Warinschi. A Computational Analysis of the Needham-Schröeder-(Lowe) Protocol. In *CSFW'03: 16th Computer Security Foundations Workshop*, pages 248–262. IEEE Computer Society, 2003.

[War05] Bogdan Warinschi. A computational analysis of the Needham-Schroeder-(Lowe) protocol. *Journal of Computer Security*, 13(3):565–591, 2005.

[Wik06] Douglas Wikström. Simplified Submission of Inputs to Protocols. Cryptology ePrint Archive, Report 2006/259, 2006.

[Wik08] Douglas Wikström. Simplified Submission of Inputs to Protocols. In *SCN'08: 6th International Conference on Security and Cryptography for Networks*, volume 5229 of *LNCS*, pages 293–308. Springer, 2008.