Replay attacks that violate ballot secrecy in Helios

Ben Smyth

Toshiba Corporation, Kawasaki, Japan

April 6, 2012

Abstract. Helios 2.0 is a web-based end-to-end verifiable electronic voting system, suitable for use in low-coercion environments. In this paper we identify a vulnerability in Helios which allows an adversary to compromise the privacy of voters whom cast abstention votes. The vulnerability can be attributed to the absence of ballot independence and the use of homomorphic ElGamal encryption, in particular, these properties can be exploited by an adversary to construct a ballot related to an abstention vote cast by an honest voter and this ballot can be submitted by a corrupt voter to influence the election outcome, thereby introducing information that can be used to violate privacy. We demonstrate the attack by breaking privacy in a mock election using the current Helios implementation. It is unlikely that the vulnerability will be exploited in a real-world election and therefore our results are largely theoretical. Nonetheless, we cannot expect any computational proofs of ballot secrecy without fixing this vulnerability and, moreover, the attack methodology may be of interest – in particular, it could represent a viable threat to existing protocols in the literature – thus providing motivation to report these results.

Keywords: Attack, ballot independence, ballot secrecy, electronic voting, Helios, homomorphic encryption, malleability, privacy.

1 Introduction

Paper-based elections derive ballot independence from physical characteristics of the real-world, for example, ballots are constructed in isolation inside polling booths and complete ballots are deposited into locked ballot boxes. By comparison, in electronic voting protocols, ballots are sent using publicly readable communication channels and, in end-to-end verifiable elections, stored on a publicly readable bulletin board; this makes the provision of ballot independence difficult in a digital setting. Nevertheless, the provision of ballot independence is important to ensure privacy, as demonstrated by Cortier & Smyth [CS11a, CS11b, SC11] who exploit the absence of ballot independence to violate ballot secrecy in the following protocols: Helios 2.0 [AMPQ09], two schemes presented

at CRYPTO – namely, the protocols due to Sako & Kilian [SK94] and Schoenmakers [Sch99] – and the scheme by Lee *et al.* [LBD+04].

Informally, ballot independence is characterised by Gennaro [Gen95, §1.1] as follows:

- *Ballot independence.* Observing another voter's interaction with the election system does not allow a voter to cast a *related* vote.

Ballot independence is important due to the apparent relationship with privacy, in addition, it 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, it prevents Bob from casting the same vote as Alice (possibly without learning Alice's vote)[1].

The concept of privacy for electronic voting systems has been informally defined by the following properties [KR05, BHM08, DKR09]:

- *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.

The above properties are ordered by increasing strength and protocols satisfying the stronger privacy properties are typically more complex, hence, a scheme satisfying ballot secrecy, rather than coercion resistance, may be preferred due its relative simplicity.

Another desirable property of electronic voting systems is verifiability [JCJ02, Dag07, KRS10], which includes the following aspects:

- *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.

The verifiability properties (also called *end-to-end verifiability* [JCJ02, CRS05, Adi06]) 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. In this paper, we revisit ballot secrecy in Helios 2.0.

Helios 2.0. Helios [AMPQ09] is an open-source web-based electronic voting system which uses homomorphic encryption. The scheme is claimed to satisfy ballot secrecy, but the nature of remote voting makes the possibility of satisfying stronger privacy properties difficult and Helios does not satisfy receipt freeness

[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 related votes whilst preserving privacy as an interesting research direction.

nor coercion resistance. In addition to ballot secrecy, the system provides individual and universal verifiability (cf. [KRS10,SRKK10] and [Smy11b, 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 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].

1.1 Foundations: Homomorphic voting schemes

An additive homomorphic encryption function E takes a key K, nonce R, and message M, and is such that

$$E(K, R, M) \circ E(K, R', M') = E(K, R + R', M + M')$$

This homomorphic property is useful in electronic voting since an encrypted tally can be derived by combining encrypted votes and ballot secrecy can be ensured using a threshold encryption scheme [CF85, BY86, SK94, Ben96, HS00]. In two candidate elections, a verifiable electronic voting protocol can be derived by coupling each encrypted vote with a signature of knowledge demonstrating that the ciphertext contains a plaintext $v \in \{0, 1\}$. These proofs are necessary to prevent an adversarial voter casting the encryption of an integer $v \notin \{0, 1\}$ which could be combined with legitimate ballots to derive an election outcome in the voter's favour, in particular, it allows the detection of ballots that could eliminate honest votes from the outcome and add multiple votes for the adversary's preferred candidate. Two candidate election systems can be generalised to multi-candidate election systems [BY86, Hir10, DJN10] by running ℓ two candidate elections in parallel and proving that the homomorphic combination of ciphertexts contains the plaintext 1. Moreover, Hirt [Hir01,Hir10] proposes a generalisation to approval voting by adding *dummy candidates*: the voter computes a ciphertext on the plaintext $v_i = 0$ (disapprove) or the plaintext $v_i = 1$ (approve) for each candidate $i \in \{1, \ldots, \ell\}$ and derives ciphertexts on the plaintexts $v_{\ell+1}, \ldots, v_{\ell+\mathsf{max}} \in \{0, 1\}$ for the dummy candidates such that $\mathsf{max} = \sum_{i=1}^{\ell+\mathsf{max}} v_i$, in addition, the voter proves that each ciphertext contains plaintext 0 or 1 and the homomorphic combination of ciphertexts contains the plaintext max, where max defines the maximum number of candidates a voter may approve. Concurrently, Damgård, Jurik & Nielsen [DJ01, DJN10] propose a similar construction using Paillier encryption. Damgård, Jurik & Nielsen also propose an optimisation to the scheme by Hirt which reduces the number of dummy ciphertexts to one: the voter computes a ciphertext c_i on the plaintext 0 or 1 for each candidate $i \in \{1, \ldots, \ell\}$, as before, and derives a ciphertext $c_{\ell+1}$ on the plaintext $v_{\ell+1} = \mathsf{max} - \sum_{i=1}^{\ell} v_i$ for the dummy candidate, in addition, the voter proves that each ciphertext $c_1, \ldots, c_\ell$ contains plaintext 0 or 1, the dummy ciphertext $c_{\ell+1}$ contains a plaintext between 0 and max, and the homomorphic combination of ciphertexts $c_1, \ldots, c_{\ell+1}$ contains the plaintext max. In Helios the dummy

candidate is removed and the voter proves that the homomorphic combination of ciphertexts contains a plaintext between 0 and max. Henceforth, we shall consider a setting where the voter must select at most one candidate and hence assume $\mathsf{max} = 1$, moreover, we shall refer to the case where a voter does not approve of any candidate as a vote for abstention. This setting can be adapted to elections where a voter must select exactly one candidate by reducing the length of the ballot to $\ell - 1$ ciphertexts and modelling votes for candidate ℓ as votes for abstention.

Exploiting privacy in schemes without independence. Cortier & Smyth [CS11a, CS11b, SC11] have shown that replaying a voter's ballot (without knowing the vote contained within that ballot) can be used to violate ballot secrecy. 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. Moreover, the vulnerability can be exploited in more realistic settings and Cortier & Smyth [CS11a, CS11b] highlight the feasibility of an attack in French legislative elections.

Variants exploiting ballot malleability. In multi-candidate elections, Cortier & Smyth propose variants of their attack which abuse the malleability of ballots to ensure replayed ballots are distinct. For example, given a valid ballot as follows:

$$\mathit{ciph}_1, \ldots, \mathit{ciph}_\ell, \mathit{spk}_1, \ldots, \mathit{spk}_\ell, \mathit{spk}$$

where ciph_i is a ciphertext containing the voter's choice for the ith candidate and spk_i demonstrates that the ciphertext ciph_i contains plaintext 0 or 1 (that is, each candidate can receive at most one vote), and spk demonstrates that the homomorphic combination of ciphertexts $\mathit{ciph}_1 \circ \cdots \circ \mathit{ciph}_\ell$ also contains plaintext 0 or 1 (that is, at most one candidate receives one vote), then the following ballot is also valid:

$$\mathit{ciph}_{\pi(1)}, \ldots, \mathit{ciph}_{\pi(\ell)}, \mathit{spk}_{\pi(1)}, \ldots, \mathit{spk}_{\pi(\ell)}, \mathit{spk}$$

where π is an arbitrary permutation over $\{1, \ldots, \ell\}$. This makes identification of replayed ballots non-trivial since checking for exact duplicates is insufficient.

At the time of writing, the aforementioned attacks by Cortier & Smyth are the only theoretical attacks against Helios. (See Adida [Adi11] for a current list of attacks.)

1.2 Contribution

This paper highlights a further attack against Helios. The attack works by exploiting the homomorphic properties of ElGamal. First, the adversary observes a ballot cast by an honest voter defined as follows:

$$\mathit{ciph}_1, \ldots, \mathit{ciph}_\ell, \mathit{spk}_1, \ldots, \mathit{spk}_\ell, \mathit{spk}$$

Secondly, the adversary computes the ciphertexts $ciph = ciph_1 \circ \cdots \circ ciph_\ell$ and $\overline{ciph} = E(K, 0, 0)$, in addition, for all $2 \leq j \leq \ell$ the adversary computes a signature of knowledge $\overline{spk}_j$ demonstrating that $\overline{ciph}$ is a ciphertext containing $v \in \{0, 1\}$. Thirdly, the adversary casts the following ballot:

$$ciph, \underbrace{\overline{ciph}, \ldots, \overline{ciph}}_{\ell-1 \text{ times}}, spk, \overline{spk}_2, \ldots, \overline{spk}_\ell, spk$$

Finally, if the election outcome contains a vote for at most one candidate (that is, the remaining candidates do not have any votes), then the aforementioned honest voter did not vote for any candidate, that is, the honest voter abstained. The attack has been demonstrated by violating privacy in a mock election using the current Helios implementation. However, the practical threat to real-world elections is small and we consider the results largely theoretical. Nevertheless, we cannot expect any computational proofs of ballot secrecy without fixing this vulnerability and, moreover, the attack methodology may be of interest – in particular, the methodology could represent a viable threat to existing protocols in the literature – thereby providing motivation to report these results.

2 Preliminaries: Helios 2.0

This section presents a full description of Helios 2.0, this background material has been largely taken from Cortier & Smyth [CS11a]. The Helios scheme exploits the additive homomorphic [CDS94, CGS97, Sch09] and distributed decryption [Ped91, CP93] properties of ElGamal [ElG85]. We will recall these cryptographic details 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 $(a, b) = (g^r \bmod p,\ g^m \cdot h^r \bmod p)$.

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 proof technique in the following section using a signature of knowledge scheme and a SHA-256 hash function denoted by $\mathcal{H}$.

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, a signature of knowledge demonstrating a disjunct 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 $\log_g g^r = \log_h (h^r \cdot g^m)/g^m$ when $m = 1$.

This technique is generalised by [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 $\mathsf{min}, \mathsf{max} \in \mathbb{N}$ such that $\mathsf{min} \leq \mathsf{max}$. Formally, a signature of knowledge demonstrating a disjunct proof of equality between discrete logarithms can be derived, and verified, as follows [AMPQ09, CDS94, Sch09].

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.
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. 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.
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 checks that the ballot does not contain a ciphertext that already exists on the bulletin board[2]. If these checks succeed, then the voter's ballot is published on the bulletin board, appended with the voter's identity id.
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.

[2] Checking ballots for ciphertexts that already exist on the bulletin board was proposed by Cortier & Smyth [CS11a, CS11b] to defend against the attacks discussed in the introduction (Section 1.1), this check is not part of the original Helios specification [AMPQ09].

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

Formally, Step 2 is defined in Figure 1. 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 Sections 2.1 & 2.2 for details).

Fig. 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 $(\bar{a}, \bar{b}, \bar{c}, \bar{s}, \bar{a}', \bar{b}', \bar{c}', \bar{s}')$.

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.
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.

3 Attacking ballot secrecy

Ballot secrecy means "a voter's vote is not revealed to anyone." We show that the Helios protocol does not satisfy this definition of ballot secrecy by presenting an attack which allows an adversary to reveal a voter's vote. Intuitively, an adversary may identify a voter's ballot on the bulletin board (using the voter's *id*) and cast a related ballot by corrupting a dishonest voter, this will leak information in the tally and the adversary can exploit this knowledge to violate the voter's privacy. A descriptions of the attack will now be presented in the case of three eligible voters.

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:

$$id_1, ciph_1, spk_1, spk'_1$$
$$id_2, ciph_2, spk_2, spk'_2$$

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

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

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).

Constructing a related ballot. The adversary observes the bulletin board and selects $ciph_k, spk_k, spk'_k$ such that id_k is the voter under attack, where $k \in \{1, 2\}$. The adversary submits the following related ballot:

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

such that for all $2 \leq i \leq \ell$ we have $\overline{spk}_i = (a_{i,0}, b_{i,0}, c_{i,0}, s_{i,0}, a_{i,1}, b_{i,1}, c_{i,1}, s_{i,1})$ where $c_{i,1}, s_{i,1}, s_{i,0} \in_R \mathbb{Z}_q^*$ and

$$\begin{aligned} a_{i,0} &= g^{s_{i,0}} \bmod p \\ a_{i,1} &= g^{s_{i,1}} \bmod p \\ b_{i,0} &= h^{s_{i,0}} \bmod p \\ b_{i,1} &= h^{s_{i,1}} \cdot g^{c_{i,1}} \bmod p \\ c_{i,0} &= \mathcal{H}(a_{i,0}, b_{i,0}, a_{i,1}, b_{i,1}) - c_{i,1} \bmod q \end{aligned}$$

It is trivial to see 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, moreover, for all $2 \leq i \leq \ell$ we have $\overline{spk}_i$ is valid proof for $(1, 1)$.

Lemma 1. *For all $2 \leq i \leq \ell$ the signature $\overline{spk}_i$ is valid for $(1, 1)$.*

Proof. Suppose $2 \leq i \leq \ell$ and $(a, b) = (1, 1)$. Let $\overline{spk}_i$ be defined above. Since $a^{c_{i,0}} = 1$ and $(b/g^0)^{c_{i,0}} = 1$, we trivially derive $g^{s_{i,0}} \equiv a_{i,0} \cdot a^{c_{i,0}} \pmod p$ and $h^{s_{i,0}} \equiv b_{i,0} \cdot (b/g^0)^{c_{i,0}} \pmod p$. Moreover, since $a^{c_{i,1}} = 1$ and $(b/g^1)^{c_{i,1}} = g^{-c_{i,1}}$, it follows that $g^{s_{i,1}} \equiv a_{i,1} \cdot a^{c_{i,1}} \pmod p$ and $h^{s_{i,1}} \equiv b_{i_1} \cdot (b/g^1)^{c_{i,1}} \pmod p$. Finally, recall $c_{i,0} = \mathcal{H}(a_{i,0}, b_{i,0}, a_{i,1}, b_{i,1}) - c_{i,1} \pmod q$ and therefore $\mathcal{H}(a_{\mathsf{min}}, b_{\mathsf{min}}, \ldots, a_{\mathsf{max}}, b_{\mathsf{max}}) \equiv c_{i,0} + c_{i,1} \pmod q$, concluding our proof.

It follows immediately that the adversary's ballot is accepted by the bulletin board, hence, we have informally shown that Helios does not satisfy ballot independence (observing another voter's interaction with the election system allows a voter to cast a *related* vote), and this will now be exploited to violate privacy.

Violating privacy. The homomorphic addition of ballots reveals the encrypted tally $(a_{1,1} \cdot a_{2,1} \cdot a_{k,1} \cdot \ldots \cdot a_{k,\ell},\ b_{1,1} \cdot b_{2,1} \cdot b_{k,1} \cdot \ldots \cdot b_{k,\ell}), (a_{1,2} \cdot a_{2,2},\ b_{1,2} \cdot b_{2,2}) \ldots, (a_{1,\ell} \cdot a_{2,\ell},\ b_{1,\ell} \cdot b_{2,\ell})$ and, given the partial decryptions, these ciphertexts can be decrypted to reveal the number of votes for each candidate. If the tally contains a vote for at most one candidate (that is, the remaining votes are for abstention), then id_k cast a vote for ϵ and hence privacy is not preserved. Moreover, the vote of the remaining honest voter will also be revealed.

Theorem 1. *If there exists r and $m \in \{0,1\}$ such that $g^m \cdot h^r = b_{1,1} \cdot b_{2,1} \cdot b_{k,1} \cdot \ldots \cdot b_{1,\ell} \cdot b_{2,\ell} \cdot b_{k,\ell}$, then the following conditions hold:*

1. *$g^0 \cdot h^{\hat{r}} = b_{k,1} \cdot \ldots \cdot b_{k,\ell}$ for some $\hat{r}$; and*
2. *$g^m \cdot h^{\hat{r}} = b_{k',i}$ for some $\hat{r}$ and $i \in \{1, \ldots, \ell\}$, where $k' \in \{1,2\} \backslash \{k\}$, and for all integers $j \in \{1, \ldots, i-1, i+1, \ldots \ell\}$ there exists $\hat{r}_j$ such that $g^0 \cdot h^{\hat{r}_j} = b_{k',j}$.*

We stress that the precondition of Theorem 1 is a trivial consequence of the tally containing at most one vote for any candidate. Postcondition (1) asserts that id_k abstained and Postcondition (2) asserts that the remaining voter voted for the candidate with one vote if $m = 1$, otherwise the remaining voter abstained. A straightforward derivative of the attack could exploit the malleability of ballots as discussed in Section 1.1. The attack has been demonstrated [Smy11a] against the current Helios implementation.

3.2 Discussion

The attack is largely theoretical since the adversary is restricted to casting one related ballot (any subsequently cast related ballots will be rejected because they contain ciphertexts that already exist on the bulletin board), accordingly, there is no motivation to generalise the attack for elections with more than three voters. Nonetheless, the results are of theoretical importance, in particular, fixing this vulnerability is a prerequsite for any computational proof of ballot secrecy.

In Appendix B we present a variant of our attack which violates the privacy of voters that cast votes for exactly one candidate, that is, voters that do not cast abstention votes. This attack has not previously been described in the literature, but would be thwarted by the additional checks – namely, checking that ballots do not contain ciphertexts that already exist on the bulletin board – proposed by Cortier & Smyth [CS11b, §5.1], nevertheless, the attack methodology may be of interest.

4 Solutions

The attack exploits a distinction between ballots cast for abstention and ballots cast for particular candidates: the homomorphic combination of ciphertexts inside a ballot for abstention contains the plaintext 0 whereas the homomorphic combination of ciphertexts inside a ballot for a particular candidate contains the plaintext 1. By comparision, in the electronic voting protocol by

Hirt [Hir01, Hir10] the dummy candidate ensures that the homomorphic combination of ciphertexts inside a ballot contains the plaintext 1 and since such a homomorphic combination cannot be used to leak information, Hirt's scheme is not vulnerable to this style of attack. Further work could consider whether the addition of a dummy candidate is sufficient for ballot secrecy in Helios, moreover, the cost of this revision could be considered; similarly, ballot secrecy could be evaluated in Hirt's scheme, since no security proof has been published. (We stress that an extension of Helios using a dummy candidate is not an instance of Hirt's scheme, because the signatures of knowledge used by these protocols are different.)

Cortier & Smyth [CS11a, CS11b] propose *ballot weeding* – namely, a ballot should not contain a ciphertext that already exists on the bulletin board – as a sufficient condition for ballot secrecy in Helios and successfully verify the security of their solution in the applied pi calculus. This analysis appears to be sound, but the model is incomplete, in particular, the homomorphic combination of an ElGamal ciphertext (a, b) with $(1, 1)$ is not captured. Omitting this detail can perhaps be justified by the requirement that "ciphertexts and signatures of knowledge should have a unique representation as *group elements*" [CS11b, §4.1] (emphasis added), however, in hindsight, modelling this detail would have been useful.

Bernhard *et al.* [BCP+11] 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, each ciphertext must be supplemented with an additional signature of knowledge, however, it is sufficient to provide a single signature of knowledge for the homomorphic combination of ciphertexts contained in a ballot; it follows that the adversary cannot successfully submit a ballot derived from the homomorphic combination of a voter's ciphertexts because the supplementary signature of knowledge cannot be constructed, therefore, the vulnerability highlighted in this paper cannot be exploited in the variant of Helios proposed by Bernhard *et al.* 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 and future work could consider whether it is sufficient to revise ballot weeding as follows: a ballot should not contain a signature of knowledge that already exists on the bulletin board (Bernhard [Ber12] and, independently, Clark [Cla12] suggest it is sufficient to check the challenges, moreover, Bernhard argues that it is necessary).

In a different direction, Gennaro [Gen95, §4.2], Cramer, Gennaro & Schoenmakers [CGS97] and Damgård, Jurik & Nielsen [DJ01, DJN10] enforce ballot independence by including the voter's identity in signatures of knowledge, and

Groth [Gro04] analyses such protocols. However, Benaloh [Ben06, Ben07] argues that the ballot encryption device should not know the voter's identity, since this information can be used to influence the behaviour of the ballot encryption device. The electronic voting protocol proposed by Juels, Catalano & Jakobsson [JCJ05] – which has been implemented by Clarkson, Chong & Myers [CCM08, CCM07] as Civitas – partially resolves this problem by binding ballots to private voter credentials, which cannot be linked to voters' identities. Moreover, their solution 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. Accordingly, alternative solutions should be sought.

5 Related work

The vulnerability highlighted in this paper is partly due to the lack of ballot independence in Helios. 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, by [CR87, 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 dependent upon the homomorphic properties of ElGamal which allow the adversary to derive a ballot related to an abstention vote cast by an honest voter, that is, the vulnerability is partly due to the possibility of constructing a ballot as a function of an honest voter's ballot. 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]. Contrary to these results, we show an attack against privacy. We believe the erroneous results reported by Langer *et al.* were due to the use of informal methods, and the approach by Volkamer & Grimm failed because only some particular scenarios were considered.

6 Conclusion

This paper identifies a vulnerability in the Helios 2.0 electronic voting protocol which can be used to violate ballot secrecy and an attack has been demonstrated against the current Helios implementation. Although the vulnerability does not pose a realistic threat to real-world elections, the results are of theoretical interest, in particular, resolving this vulnerability is a prerequisite to proving ballot secrecy in the computational model. In addition, the attack methodology may be of interest, since it could represent a viable threat against existing protocols in the literature.

Acknowledgements

I am grateful to David Bernhard, Jeremy Clark and Olivier Pereira for their careful reading of draft versions of this paper and subsequent discussions; their feedback helped improve the paper.

A Signatures of knowledge

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

A.1 Knowledge of discrete logs

Given the aforementioned cryptographic parameters (p, q, g), 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, $h \equiv g^x \bmod p$.

A.2 Equality between discrete logs

Given the aforementioned cryptographic parameters (p, q, g), 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 $k \equiv g^x \bmod p$.

For our purposes, 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; that is, the proof asserts $\log_g h_i = \log_a k_i$, as required for integrity of decryption.

B A variant of our attack

This appendix presents a variant of the attack described in Section 3 which violates the privacy of voters in the original Helios scheme, but is thwarted by the solution proposed by Cortier & Smyth [CS11b, §4.1]. Let us suppose the bulletin board is given in Section 3.1 and the adversary selects $ciph_k, spk_k, spk_k'$ such that id_k is the voter under attack, where $k \in \{1, 2\}$. The attack proceeds as follows.

Constructing a related ballot. The adversary selects integer $v \in \{1, \ldots, \ell\}$ and submits the following related ballot:

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

where $\overline{spk}_i$ is given in Section 3.1 for all $i \in \{1, \ldots, v-1, v+1, \ldots, \ell\}$. It is trivial to see that $spk_{k,v}$ is a valid proof for $(a_{k,v}, b_{k,v})$ and, by Lemma 1, we have $\overline{spk}_i$ is valid proof for $(1, 1)$ for all $i \in \{1, \ldots, v-1, v+1, \ldots, \ell\}$. Moreoever, $spk_{k,v}$ is

a valid proof for the homomorphic combination of all the ciphertexts contained in the adversary's ballot. It follows that the adversary's ballot will be accepted by the bulletin board, hence, we have informally shown another technique to violate ballot independence in Helios.

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

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

Given the partial decryptions, the tally can be decrypted to reveal the number of votes for each candidate. If the tally contains at least two votes for a candidate, then id_k cast a vote for that candidate and hence privacy is not preserved. Moreover, the vote of the remaining honest voter will also be revealed.

Theorem 2. *If there exists r and $m \in \{2, 3\}$ such that $g^m \cdot h^r \in \{B_1, \ldots, B_\ell\}$, then the following conditions hold:*

1. *$b_{k,v} = g^1 \cdot h^{\hat{r}}$ for some $\hat{r}$; and*
2. *$b_{k',j} = g^1 \cdot h^{\hat{r}}$ for some $\hat{r}$ and $j \in \{1, \ldots, \ell\}$, where $k' \in \{1, 2\} \backslash \{k\}$.*

The precondition of Theorem 2 is an immediate consequence of the tally containing at least two votes for a candidate. Postcondition (1) asserts that id_k voted for candidate t_v and Postcondition (2) asserts that the remaining voter voted for candidate t_j.

References

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

[Adi11] Ben Adida. Attacks and Defenses. Helios documentation, `http://documentation.heliosvoting.org/attacks-and-defenses` (accessed 8 December 2011), 2011.

[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.

[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.

[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, 15th 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.

[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.

[CF85] Josh Daniel Cohen and Michael J. Fischer. A Robust and Verifiable Cryptographically Secure Election Scheme. In *FOCS'85: 26th Symposium on Foundations of Computer Science*, pages 372–382. IEEE Computer Society, 1985.

[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 *EURO-*

CRYPT'97: 16th International Conference on the Theory and Applications of Cryptographic Techniques, volume 1233 of *LNCS*, pages 103–118. Springer, 1997.

[Cla12] Jeremy Clark. Private email communication, 4th April 2012.

[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.

[CS11a] Véronique Cortier and Ben Smyth. Attacking and fixing helios: An analysis of ballot secrecy. Cryptology ePrint Archive, Report 2010/625 (version 20111110:012334), 2011.

[CS11b] 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/`.

[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.

[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.

[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.

[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.

[Gro04] Jens Groth. Evaluating Security of Voting Schemes in the Universal Composability Framework. In *ACNS'04: 2nd International Conference on Applied Cryptography and Network Security*, volume 3089 of *LNCS*, pages 46–60. Springer, 2004.

[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.

[Hir01] Martin Hirt. *Multi-Party Computation: Efficient Protocols, General Adversaries, and Voting*. PhD thesis, ETH Zurich, 2001.

[Hir10] Martin Hirt. Receipt-Free K-out-of-L Voting Based on ElGamal Encryption. 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 64–82. Springer, 2010.

[HS00] Martin Hirt and Kazue Sako. Efficient Receipt-Free Voting Based on Homomorphic Encryption. In *EUROCRYPT'06: 25th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 1807 of *LNCS*, pages 539–556. Springer, 2000.

[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`.

[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.

[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.

[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.

[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.

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

[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.

[Smy11a] `https://vote.heliosvoting.org/helios/elections/3016bac6-2224-11e1-b02d-12313f028a58/view` (accessed 12 December 2011), 2011.

[Smy11b] Ben Smyth. *Formal verification of cryptographic protocols with automated reasoning*. PhD thesis, School of Computer Science, University of Birmingham, 2011.

[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.

[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.