Non-malleable encryption with proofs of plaintext knowledge and applications to voting

Ben Smyth[1] and Yoshikazu Hanatani[2]

[1]Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg
[2]Toshiba Corporation, Kawasaki, Japan

December 20, 2018

Abstract

Non-malleable asymmetric encryption schemes which prove plaintext knowledge are sufficient for secrecy in some domains. For example, ballot secrecy in voting. In these domains, some applications derive encryption schemes by coupling malleable ciphertexts with proofs of plaintext knowledge, without evidence that the sufficient condition (for secrecy) is satisfied nor an independent security proof (of secrecy). Consequently, it is unknown whether these applications satisfy desirable secrecy properties. In this article, we propose a generic construction for such a coupling and show that our construction produces non-malleable encryption schemes which prove plaintext knowledge. Furthermore, we show how our results can be used to prove ballot secrecy of voting systems. Accordingly, we facilitate the development of applications satisfying their security objectives.

1 Introduction

An additively-homomorphic asymmetric encryption scheme allows a pair of enciphered plaintexts to be combined into a single ciphertext, such that the single ciphertext enciphers the sum of those plaintexts. Thus, homomorphic encryption allows aggregation without decryption, which is useful in many domains, including social choice theory. For example, a two-candidate voting system might instruct voters to cast asymmetric encryptions of their votes “yes” (0) or “no” (1), and instruct the tallier to decrypt the combination of encrypted votes to reveal the number of no-votes, from which the number of yes-votes can be deduced. This system ensures ballot secrecy (when instantiated with a suitable encryption scheme), because encryption prevents votes being recovered from ballots and the tallying procedure ensures that individual votes are not revealed.[1] However, talliers and voters may cheat. Indeed, a malicious tallier many claim a spurious number of yes- and no-votes, or discard some ciphertexts under the guise that they are ill-formed (possibly after peaking inside). Moreover, a malicious voter might encrypt a negative integer to switch no-votes to yes-votes, or an integer greater than one to cast multiple no-votes. To prevent cheating, ciphertexts can be coupled with non-interactive proofs demonstrating correct ciphertext construction (to prevent claims of ill-formedness) for plaintext 0 or 1 (to prevent adversarial voters switching yes- and no-votes).

Smyth shows that building voting systems from non-malleable encryption schemes suffices for ballot secrecy [64], moreover, Quaglia & Smyth show that such systems can be transformed into auction systems satisfying bid secrecy [57]. Furthermore, Bernhard, Pereira & Warinschi show that coupling an IND-CPA encryption scheme with a non-interactive zero-knowledge proof suffices to achieve non-malleable encryption [10]. It follows that our exemplar voting system satisfies ballot secrecy (when non-interactive zero-knowledge proofs are used and when the underlying encryption scheme satisfies IND-CPA). Unfortunately, results by Bernhard, Pereira & Warinschi do not apply to many encryption schemes that are used to construct voting systems, such as those proposed by Hirt [42, 43], Damgård, Jurik & Nielsen [27, 28], and Adida *et al.* [1], as we shall now discuss.

1.1 Related work: Encryption schemes for voting systems and their shortcomings

Hirt [42, 43] proposes a construction for encryption schemes, from schemes satisfying IND-CPA, for a block of messages $m_1, \ldots, m_k \in \{0,1\}$, such that the homomorphic combination of messages in the block is between 1 and max: A ciphertext c_i is generated on message m_i for $1 \leq i \leq k$, and ciphertexts $c_{k+1}, \ldots, c_{k+\mathsf{max}}$ are generated on *dummy messages* $m_{k+1}, \ldots, m_{k+\mathsf{max}} \in \{0,1\}$ such that $\mathsf{max} = \sum_{j=1}^{k+\mathsf{max}} m_j$, in addition, proofs of knowledge are used to demonstrate that each ciphertext contains plaintext 0 or 1 and the homomorphic combination of ciphertexts $c_1, \ldots, c_{k+\mathsf{max}}$ contains plaintext max, where non-interactive proofs use a common challenge derived from the ciphertexts and commitments. Concurrently, Damgård, Jurik & Nielsen [27, 28] propose a similar construction using Paillier encryption, but their work is reliant on unique identifiers to achieve non-malleability and it is unclear what security guarantees can be achieved when state is not maintained. Damgård, Jurik & Nielsen also propose an optimisation to the scheme by Hirt which reduces the number of dummy ciphertexts to one: A ciphertext c_i is generated on each message m_i for $1 \leq i \leq k$, as before, and a ciphertext c_{k+1} is generated on

[1] Ballot secrecy necessarily assumes that the tallier does not deviate from the prescribed tallying procedure, since ballots can be tallied individually to reveal votes. Distributing the tallier's role permits ballot secrecy under the weaker assumption that at least one tallier does not deviate, but a trust assumption nonetheless remains. Ultimately, we would prefer not to trust talliers; unfortunately, this is only known to be possible for decentralised voting systems, e.g., [62, 47, 38, 41, 45, 46], which do not scale.

plaintext $m_{k+1} = \mathsf{max} - \sum_{i=1}^{k} m_i$ for the dummy candidate, in addition, proofs of knowledge are used to demonstrate that the ciphertexts $c_1, \ldots, c_k$ contain plaintext 0 or 1, the dummy ciphertext c_{k+1} contains a plaintext between 0 and max, and the homomorphic combination of ciphertexts $c_1, \ldots, c_{k+1}$ contains plaintext max. Adida *et al.* [1] generalise Hirt's scheme to consider cases where the homomorphic combination of messages $m_1, \ldots, m_k$ is between min and max, this is achieved by removing the dummy ciphertexts and proving that the homomorphic combination of ciphertexts $c_1, \ldots, c_k$ contains a plaintext between min and max (the non-interactive proofs proposed by Adida *et al.* do not include ciphertexts in challenges).

Adida *et al.* use their scheme instantiated with El Gamal to build the Helios voting system [1], which instructs each voter to select their vote v from candidates $1, \ldots, nc$ and compute ciphertexts $c_1, \ldots, c_{nc-1}$ such that if $v \neq nc$, then ciphertext c_v contains plaintext 1 and the remaining ciphertexts contain plaintext 0, otherwise, all ciphertexts contain plaintext 0. (Only $nc - 1$ ciphertexts are needed, rather than nc, because a vote for candidate nc is uniquely represented when all ciphertexts contain plaintext 0. As is the case for our exemplar yes-no voting system, described in the opening paragraph.) Moreover, the voter computes zero-knowledge proofs $\sigma_1, \ldots, \sigma_{nc}$ demonstrating correct computation. Proof σ_j demonstrates that ciphertext c_j contains 0 or 1, where $1 \leq j \leq nc - 1$, and proof σ_{nc} demonstrates that the homomorphic combination of ciphertexts $c_1 \otimes \cdots \otimes c_{nc-1}$ contains 0 or 1. The voter casts those El Gamal ciphertexts and proofs as their ballot, which is a ciphertext in the encryption scheme proposed by Adida *et al.* [1]. It follows from results by Bernhard, Pereira & Warinschi that Helios uses non-malleable encryption for two candidate elections [10]. Moreover, Helios satisfies ballot secrecy for such two candidate elections. However, Cortier & Smyth show that the encryption scheme by Adida *et al.* is malleable in the general case [21, 22]. Indeed, given (an Adida *et al.*) ciphertext $c_1, \ldots, c_{nc-1}, \sigma_1, \ldots, \sigma_{nc}$, we have $c_{\chi(1)}, \ldots, c_{\chi(nc-1)}, \sigma_{\chi(1)}, \ldots, \sigma_{\chi(nc-1)}, \sigma_{nc}$ is a ciphertext for all permutations χ on $\{1, \ldots, nc - 1\}$, hence, the encryption scheme proposed by Adida *et al.* [1] is malleable. It follows that Helios does not satisfy ballot secrecy in the general case. Moreover, Bernhard, Pereira & Warinschi show that the proof system used by Adida *et al.* belongs to the class of *weak Fiat-Shamir transformations* and demonstrate additional issues. Furthermore, Smyth shows that the forthcoming Helios release (that uses the Fiat-Shamir transformation, rather than its weak variant) does not satisfy ballot secrecy due to the absence of non-malleability [64].

We build upon results by Adida *et al.* and Bernhard, Pereira & Warinschi to derive a non-malleable asymmetric encryption scheme suitable for voting systems, auction systems, and other systems with secrecy and verifiability requirements.

1.2 Contribution, structure, & context

Section 2 proposes a generic construction for non-malleable asymmetric encryption schemes on blocks of plaintexts from homomorphic encryption schemes and proofs of knowledge. Our construction is inspired by Adida *et al.* [1]; it differs by including ciphertexts and block numbers in challenges, to prevent attacks. Section 3 discusses sufficient conditions for ballot secrecy and shows how our results give way to ballot-secrecy proofs for voting systems, including a variant of the Helios voting system (the original Helios system is insecure). The remaining sections present further related work (§4) and a brief conclusion (§5). The appendices recall cryptographic primitives and their associated security properties (Appendix A), along with a formal definition of Helios (Appendix B), and present proofs (Appendix C).

2 Non-malleable encryption with proofs of plaintext knowledge

We present our construction for non-malleable asymmetric encryption schemes on blocks of plaintexts from homomorphic encryption schemes and proofs of plaintext knowledge in a subspace as follows. (We recall definitions of an *asymmetric encryption scheme*, *homomorphic encryption*, and a *non-interactive proof system*, along with security definitions, in Appendix A. We also recall definitions of a sigma protocol which *proves plaintext knowledge in a subspace* and of the *Fiat-Shamir transformation*, the former also defines a *subspace*.)

Definition 1. *Let* $\Pi = (\mathsf{Gen}_\Pi, \mathsf{Enc}_\Pi, \mathsf{Dec}_\Pi)$ *be a homomorphic asymmetric encryption scheme (with respect to ternary operators* $\odot$*,* $\oplus$*, and* $\otimes$*),* $\Delta = (\mathsf{Prove}, \mathsf{Verify})$ *be a non-interactive proof system derived by application of the Fiat-Shamir transformation to a hash function and a sigma protocol that proves plaintext knowledge in a subspace (of the encryption scheme's message space), and* ℓ *be a positive integer. We define* $\gamma(\Pi, \Delta, \ell) = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *as follows:*

- $\mathsf{Gen}(\kappa)$ *computes* $(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}_\Pi(\kappa); \mathfrak{m} \leftarrow \{(m_1, \ldots, m_\ell) \mid m_1, \ldots, m_\ell \in \mathfrak{M} \wedge m_1 \odot \cdots \odot m_\ell \in \mathfrak{M}\}$ *and outputs* $(pk, sk, \mathfrak{m})$*, where* $\mathfrak{M}$ *is the subspace.*

- $\mathsf{Enc}(pk, m)$ *parses* m *as a block of plaintexts* $(m_1, \ldots, m_\ell)$*, chooses coins* $r_1, \ldots, r_\ell$ *uniformly at random, computes*

  for $1 \leq i \leq \ell$ **do**
  $\quad c_i \leftarrow \mathsf{Enc}_\Pi(pk, m_i; r_i);$
  $\quad \sigma_i \leftarrow \mathsf{Prove}((pk, c_i, \mathfrak{M}), (m_i, r_i), i, \kappa);$
  $\sigma \leftarrow \mathsf{Prove}((pk, c_1 \otimes \cdots \otimes c_\ell, \mathfrak{M}), (m_1 \odot \cdots \odot m_\ell, r_1 \oplus \cdots \oplus r_\ell), \ell + 1, \kappa);$

  and outputs $(c_1, \sigma_1, \ldots, c_\ell, \sigma_\ell, \sigma)$*.*

- $\mathsf{Dec}(sk, c)$ *parses* c *as* $(c_1, \sigma_1, \ldots, c_\ell, \sigma_\ell, \sigma)$ *and outputs* $(\mathsf{Dec}_\Pi(sk, c_1), \ldots, \mathsf{Dec}_\Pi(sk, c_\ell))$ *if parsing succeeds and* $\mathsf{Verify}((pk, c_1 \otimes \cdots \otimes c_\ell, \mathfrak{M}), \sigma, \ell + 1, \kappa) \wedge \bigwedge_{1 \leq i \leq \ell} \mathsf{Verify}((pk, c_i, \mathfrak{M}), \sigma_i, i, \kappa)$*, and outputs* $\perp$ *otherwise.*

In the above construction, observe that the message space is restricted such that for all ciphertexts $(c_1, \sigma_1, \ldots, c_\ell, \sigma_\ell, \sigma)$ output by $\mathsf{Enc}(pk, (m_1, \ldots, m_\ell))$, we have $((pk, c_i, \mathfrak{M}), (m_1, r_i))$ is an element of the non-interactive proof system's relation, where $1 \leq i \leq \ell$ and r_i are the coins used to construct c_i. Moreover, $((pk, c_1 \otimes \cdots \otimes c_\ell, \mathfrak{M}), (m_1 \odot \cdots \odot m_\ell, r_1 \oplus \cdots \oplus r_\ell))$ is an element of the relation too. Hence, it follows – by completeness of the non-interactive proof system – that schemes generated using our construction satisfy the correctness property of asymmetric encryption schemes. That is, γ constructs asymmetric encryption schemes.

Lemma 1 (Correctness)**.** *Given a homomorphic asymmetric encryption scheme* Π*, a non-interactive proof system* Δ *derived by application of the Fiat-Shamir transformation to a hash function and a sigma protocol that proves plaintext knowledge in a subspace, and a positive integer* ℓ*, we have* $\gamma(\Pi, \Delta, \ell)$ *is an asymmetric encryption scheme.*

Our proof of Lemma 1 and all further proofs appear in Appendix C.

Our construction builds upon homomorphic asymmetric encryption schemes and proofs of plaintext knowledge in a subspace to enhance functionality of the resulting encryption scheme. For example, homomorphic operations can be performed on the malleable ciphertexts encapsulated in a non-malleable ciphertext. That is, given a non-malleable ciphertext $(c_1, \sigma_1, , \ldots, c_\ell, \sigma_\ell, \sigma)$, homomorphic operations can be performed on the encapsulated ciphertexts $c_1, \ldots, c_\ell$. Moreover, if proofs $\sigma_1, \ldots, \sigma_\ell$ are valid, then each of those ciphertexts contain plaintexts in the subspace. Furthermore, the homomorphic combination of ciphertexts, namely, $c_1 \otimes \cdots \otimes c_\ell$, contains a plaintext in the subspace if proof σ is valid. This enhanced functionality justifies the efficiency cost incurred from non-interactive proofs. Indeed, asymmetric encryption schemes derived using our construction are useful for privacy preserving applications and, in Section 3, we will demonstrate the applicability of our results in the context of voting systems. First, we prove that our construction produces schemes satisfying comparison based non-malleability under chosen plaintext attack (CNM-CPA) [4].

Theorem 2. *Let* Π *be a homomorphic asymmetric encryption scheme,* Δ *be a non-interactive proof system derived by application of the Fiat-Shamir transformation to a random oracle and a sigma protocol that proves plaintext knowledge in a subspace, and* ℓ *be a positive integer. If* Π *satisfies* IND-CPA *and is perfectly correct, and the sigma protocol satisfies special soundness and special honest verifier zero-knowledge, then asymmetric encryption scheme* $\gamma(\Pi, \Delta, \ell)$ *satisfies* CNM-CPA*.*

Bellare & Sahai have shown that CNM-CPA is equivalent to indistinguishability under a parallel chosen-ciphertext attack (IND-PA0) [4], hence, our theorem can be equivalently stated in terms of indistinguishability.

3 Applications to voting

An election is a decision-making procedure to choose representatives [48, 59, 40, 2]. Choices should be made freely, and this must be ensured by voting systems [71, 53, 54]. Many voting systems rely on art, rather than science, to ensure that choices are made freely. Such systems build upon creativity and skill, rather than scientific foundations, and are routinely broken in ways that compromise free choice, e.g., [37, 13, 75, 76, 70]. Breaks can be avoided by proving that systems satisfy carefully formulated security definitions that capture voters voting freely. We use such a definition to analyse Helios.

3.1 Election schemes

We consider the class of voting systems that consist of the following four steps. First, a tallier generates a key pair. Secondly, each voter constructs and casts a ballot for their preferred candidate. Thirdly, the tallier tallies the ballots and announces a distribution of candidate preferences. Finally, voters and other interested parties check that the distribution corresponds to preferences expressed in ballots. Such systems can be formally captured by the following *election scheme* syntax proposed by Smyth, Frink & Clarkson [68].

Definition 2 (Election scheme [68]). *An* election scheme *is a tuple of probabilistic polynomial-time algorithms* $(\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ *such that:*

Setup, *denoted* $(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa)$, *is run by the tallier. The algorithm takes a security parameter* κ *as input and outputs a key pair* pk, sk, *a maximum number of ballots* mb, *and a maximum number of candidates* mc.

Vote, *denoted* $b \leftarrow \mathsf{Vote}(pk, v, nc, \kappa)$, *is run by voters. The algorithm takes as input a public key* pk, *a voter's vote* v, *some number of candidates* nc, *and a security parameter* κ. *The vote should be selected from a sequence* $1, \ldots, nc$ *of candidates. The algorithm outputs a ballot* b *or error symbol* $\perp$.

Tally, *denoted* $(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, nc, \kappa)$, *is run by the tallier. The algorithm takes as input a private key* sk, *a bulletin board* $\mathfrak{bb}$, *some number of candidates* nc, *and a security parameter* κ, *where* $\mathfrak{bb}$ *is a set. And outputs an election outcome* $\mathfrak{v}$ *and a non-interactive tallying proof* pf *demonstrating that the outcome corresponds to votes expressed in ballots on the bulletin board. The election outcome* $\mathfrak{v}$ *should be a vector of length* nc *such that* $\mathfrak{v}[v]$ *indicates the number of votes for candidate* v.

Verify, *denoted* $s \leftarrow \mathsf{Verify}(pk, \mathfrak{bb}, nc, \mathfrak{v}, pf, \kappa)$, *is run to audit an election. The algorithm takes as input a public key* pk, *a bulletin board* $\mathfrak{bb}$, *some number of candidates* nc, *an election outcome* $\mathfrak{v}$, *a tallying proof* pf, *and a security parameter* κ. *And outputs a bit* s, *which is* 1 *if the election verifies successfully or* 0 *otherwise.*

Election schemes must satisfy correctness*: there exists a negligible function* negl*, such that for all security parameters* κ*, integers* nb *and* nc*, and votes* $v_1, \ldots, v_{nb} \in \{1, \ldots, nc\}$*, it holds that, given a zero-filled vector* $\mathfrak{v}$ *of length* nc*, we have:* $\Pr[(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa);$ $\mathbf{for}\ 1 \leq i \leq nb\ \mathbf{do}\ \{\ b_i \leftarrow \mathsf{Vote}(pk, v_i, nc, \kappa); \mathfrak{v}[v_i] \leftarrow \mathfrak{v}[v_i] + 1\ \}\ (\mathfrak{v}', pf) \leftarrow \mathsf{Tally}(sk, \{b_1, \ldots, b_{nb}\}, nc, \kappa) : nb \leq mb \wedge nc \leq mc \Rightarrow \mathfrak{v} = \mathfrak{v}'] > 1 - \mathsf{negl}(\kappa)$.

3.2 Sufficient conditions for ballot secrecy

Smyth [64] defines *ballot secrecy* (Ballot-Secrecy) to capture a notion of free-choice and shows that ballot secrecy coincides with *ballot independence* (IND-CVA) when proofs output during tallying are zero-knowledge and when honestly constructed ballots are correctly tallied (HB-Tally-Soundness). Moreover, he shows that HB-Tally-Soundness is implied by verifiability. Hence, to prove ballot secrecy for verifiable election schemes, it suffices to prove ballot independence, assuming tallying proofs are zero-knowledge.

Definition 3 (IND-CVA [64])**.** *Let* $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ *be an election scheme,* $\mathcal{A}$ *be an adversary,* κ *be the security parameter, and* $\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{A}, \kappa)$ *be the following game.*

$\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{A}, \kappa) =$

$(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa);$
$(v_0, v_1, nc) \leftarrow \mathcal{A}(pk, \kappa);$
$\beta \leftarrow_R \{0, 1\};$
$b \leftarrow \mathsf{Vote}(pk, v_\beta, nc, \kappa);$
$\mathfrak{bb} \leftarrow \mathcal{A}(b);$
$(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, nc, \kappa);$
$g \leftarrow \mathcal{A}(\mathfrak{v});$
$\mathbf{return}\ g = \beta \wedge b \notin \mathfrak{bb} \wedge 1 \leq v_0, v_1 \leq nc \leq mc \wedge |\mathfrak{bb}| \leq mb;$

We say Γ *satisfies* ballot independence *or* indistinguishability under chosen vote attack *(*IND-CVA*), if for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, there exists a negligible function* negl*, such that for all security parameters* κ*, we have* $\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{A}, \kappa) \leq \frac{1}{2} + \mathsf{negl}(\kappa)$.

Definition 4 (Zero-knowledge tallying proofs [64])**.** *Let* $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ *be an election scheme. We say* Γ *has* zero-knowledge tallying proofs*, if there exists a zero-knowledge non-interactive proof system* (Prove, Verify)*, such that for all security parameters* κ*, integers* nc*, bulletin boards* $\mathfrak{bb}$*, outputs* (pk, sk, mb, mc) *of* $\mathsf{Setup}(\kappa)$*, and outputs* $(\mathfrak{v}, pf)$ *of* $\mathsf{Tally}(sk, \mathfrak{bb}, nc, \kappa)$*, we have* $pf = \mathsf{Prove}((pk, \mathfrak{bb}, nc, \mathfrak{v}), sk, \kappa; r)$*, such that coins* r *are chosen uniformly at random by* Tally*.*

Theorem 3. *Let* Γ *be an election scheme with zero-knowledge tallying proofs. Suppose* Γ *satisfies* HB-Tally-Soundness*. We have* Γ *satisfies* Ballot-Secrecy *iff* Γ *satisfies* IND-CVA*.*

A proof of Theorem 3 appears in [64]. We exploit the theorem to simplify analysis of ballot secrecy in Helios.

Suitability of Ballot-Secrecy. Discussion of ballot secrecy originates from Chaum [18] and the earliest definitions of ballot secrecy are due to Benaloh *et al.* [6, 7, 5].[2] More recently, Bernhard *et al.* propose a series of ballot secrecy definitions [10, 66, 67, 8]. Smyth shows that these definitions do not detect vulnerabilities that arise when an adversary controls the bulletin board or the communication channel [64]. By comparison, the definition of ballot secrecy that we consider (Definition 3) detects such vulnerabilities and appears to be the strongest definition in the literature.

3.3 Case study: Helios

Helios can be informally modelled as an election scheme such that:

Setup generates a key pair for an asymmetric homomorphic encryption scheme, proves correct key generation in zero-knowledge, and outputs the key pair along with the proof.

Vote enciphers the vote to a ciphertext, proves in zero-knowledge that the ciphertext is correctly constructed and that the vote is selected from the sequence of candidates, and outputs the ciphertext coupled with the proof.

Tally selects the ballots on the bulletin board for which proofs hold, homomorphically combines the ciphertexts in those ballots, decrypts the homomorphic combination to reveal the election outcome, and announces the outcome, along with a zero-knowledge proof of correct decryption.

Verify checks the proofs and accepts the outcome if these checks succeed.

Helios was first released in 2009 as *Helios 2.0*, the current release is *Helios 3.1.4*, and a new release is planned.[3] Henceforth, we'll refer to the planned release as *Helios'12*.

Smyth proves that neither Helios 2.0 nor Helios 3.1.4 satisfy Ballot-Secrecy [64]. Moreover, he reasons that Helios'12 does not satisfy Ballot-Secrecy either. This is due to the use of malleable ballots in Helios 2.0, Helios 3.1.4 & Helios'12.[4] Smyth, Frink & Clarkson propose a generic construction Helios for Helios-like election schemes, which is parameterised on the choice of homomorphic encryption scheme and sigma protocols. That construction can be used to derive a variant of Helios called *Helios'16* [68]. (We formally define Helios and Helios'16 in Appendix B.) We use our results to prove that Helios'16 satisfies ballot secrecy.

---

[2] Quaglia & Smyth present a tutorial-style introduction to modelling ballot secrecy [58], and Smyth provides a technical introduction [65].

[3] `http://documentation.heliosvoting.org/verification-specs/helios-v4`, published c. 2012, accessed 18 Oct 2017. (Cached version: `https://web.archive.org/web/*/https://documentation.heliosvoting.org/verification-specs/helios-v4`.)

[4] Helios'12 uses non-malleable ballots for two candidate elections and is proven to satisfy notions of ballot secrecy [10, 8], assuming the bulletin board and the communication channel are trusted. (See Smyth [64] for further details.)

Theorem 4. *Helios'16 satisfies* Ballot-Secrecy.

A proof of Theorem 4 appears in Appendix C.3. Smyth also presents a proof [64]. These proofs are different in structure. In particular, we exploit Theorem 2 to simplify our proof, which allows us to present our proof in just two pages, whereas Smyth required five.

4 Further related work

In complimentary work, constructions for IND-CCA1 and IND-CCA2 secure encryption schemes from schemes satisfying IND-CPA have been presented. Naor & Yung [51] propose a construction for IND-CCA1 encryption schemes from IND-CPA schemes by encrypting each plaintext twice and using a proof of knowledge to demonstrate that both ciphertexts encrypt the same plaintext, moreover, Sahai [60] defines additional conditions to achieve IND-CCA2 security and efficient variants which maintain IND-CCA2 security are known, e.g., [49, 9]. Cramer & Shoup [26] derive IND-CCA2 secure encryption schemes from IND-CPA schemes using *"universal hash proof systems,"* furthermore, efficient constructions have been shown, e.g., [25, 26, 29]. Elkind & Sahai [32] have shown that the techniques by Naor & Yung and Cramer & Shoup are special cases of a more general paradigm: given an IND-CPA secure encryption scheme (where *"ill-formed"* ciphertexts are indistinguishable from *"well-formed"* ciphertexts), an IND-CCA2 secure encryption scheme can be constructed by coupling ciphertexts with *"proofs of well-formedness."* Fujisaki & Okamoto [34] derive IND-CCA2 secure encryption schemes from probabilistic trapdoor one-way functions satisfying IND-CPA (such as El Gamal). Canetti, Halevi & Katz [15] derive IND-CCA2 encryption schemes from IND-CPA secure identity-based encryption schemes and more efficient variants are known, e.g., [12, 14, 11]. These results are orthogonal to our work, since we explicitly focus on schemes using proofs of knowledge to achieve additional security properties. The aforementioned schemes do not.

Discussion of ballot independence originates from Gennaro [35] and the relationship with ballot secrecy has been explored: Benaloh shows that a simplified version of his voting system allows the administrator's private key to be recovered by an adversary who casts a ballot as a function of other voters' ballots [5, §2.9] and, more generally, Sako & Kilian [61, §2.4], Michels & Horster [50, §3], Wikström [72, 73, 74] and Cortier & Smyth [22, 21] discuss how malleable ballots can be abused to compromise ballot secrecy. The first definition of ballot independence seems to be due to Smyth & Bernhard [66, 67], moreover, they formally prove relations between their definitions of secrecy and independence. Independence has also been studied beyond elections, e.g., [19], and the possibility of compromising security in the absence of independence has been considered, e.g., [20, 56, 55, 30, 31, 36].

An earlier version of our construction appeared in [69]. In that work, we considered a construction from a sigma protocol, rather than a non-interactive proof system derived from a sigma protocol and a hash function. This overcomplicated the construction and proofs. Thus, this work improves upon our earlier work by considerably simplifying results. Moreover, we have used our results to prove that a variant of Helios satisfies ballot secrecy, whereas our earlier work did not include such a proof.

5 Conclusion

We deliver a generic construction for non-malleable asymmetric encryption schemes from homomorphic schemes coupled with proofs of knowledge. The proofs of knowledge enhance functionality. In particular, an observer can check that ciphertexts contain plaintexts from a particular message space, without the private key. Moreover, homomorphic operations can be performed on the malleable ciphertexts embedded in non-malleable ciphertexts. We believe that the enhanced functionality justifies the efficiency cost. Indeed, such functionality has proven to be particularly useful in secret, verifiable voting systems.

A Cryptographic primitives

A.1 Asymmetric encryption

Definition 5 (Asymmetric encryption scheme [44])**.** *An* asymmetric encryption scheme *is a tuple of probabilistic polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, such that:*[5]

- **Gen***, denoted* $(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(\kappa)$*, inputs a security parameter* κ *and outputs a key pair* (pk, sk) *and message space* $\mathfrak{m}$*.*
- **Enc***, denoted* $c \leftarrow \mathsf{Enc}(pk, m)$*, inputs a public key pk and message* $m \in \mathfrak{m}$*, and outputs a ciphertext c.*
- **Dec***, denoted* $m \leftarrow \mathsf{Dec}(sk, c)$*, inputs a private key sk and ciphertext c, and outputs a message m or an error symbol. We assume* Dec *is deterministic.*

Moreover, the scheme must be correct*: there exists a negligible function* negl*, such that for all security parameters* κ *and messages m, we have* $\Pr[(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(\kappa); c \leftarrow \mathsf{Enc}(pk, m) : m \in \mathfrak{m} \Rightarrow \mathsf{Dec}(sk, c) = m] > 1 - \mathsf{negl}(\kappa)$*. A scheme has* perfect correctness *if the probability is* 1*.*

Definition 6 (Homomorphic encryption [68])**.** *An asymmetric encryption scheme* $\Gamma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* homomorphic*, with respect to ternary operators* $\odot$*,* $\oplus$*,*

[5] Our definition differs from Katz and Lindell's original definition [44, Definition 10.1] in that we formally state the plaintext space.

and $\otimes$,[6] *if there exists a negligible function* negl*, such that for all security parameters* κ*, we have the following.*[7] *First, for all messages* m_1 *and* m_2 *we have* $\Pr[(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(\kappa); c_1 \leftarrow \mathsf{Enc}(pk, m_1); c_2 \leftarrow \mathsf{Enc}(pk, m_2) : m_1, m_2 \in \mathfrak{m} \Rightarrow \mathsf{Dec}(sk, c_1 \otimes_{pk} c_2) = \mathsf{Dec}(sk, c_1) \odot_{pk} \mathsf{Dec}(sk, c_2)] > 1 - \mathsf{negl}(\kappa)$*. Secondly, for all messages* m_1 *and* m_2*, and all coins* r_1 *and* r_2*, we have* $\Pr[(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(\kappa) : m_1, m_2 \in \mathfrak{m} \Rightarrow \mathsf{Enc}(pk, m_1; r_1) \otimes_{pk} \mathsf{Enc}(pk, m_2; r_2) = \mathsf{Enc}(pk, m_1 \odot_{pk} m_2; r_1 \oplus_{pk} r_2)] > 1 - \mathsf{negl}(\kappa)$*. We say* Γ *is* additively homomorphic*, if for all security parameters* κ*, key pairs* pk, sk*, and message spaces* $\mathfrak{m}$*, such that there exists coins* r *and* $(pk, sk, \mathfrak{m}) = \mathsf{Gen}(\kappa; r)$*, we have* $\odot_{pk}$ *is the addition operator in group* $(\mathfrak{m}, \odot_{pk})$*.*

Definition 7 (IND-CPA [3])**.** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an asymmetric encryption scheme,* $\mathcal{A}$ *be an adversary,* κ *be the security parameter, and* $\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \kappa)$ *be the following game.*[8]

$\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \kappa) =$
 $(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(\kappa);$
 $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathfrak{m}, \kappa);$
 $\beta \leftarrow_R \{0, 1\};$
 $c \leftarrow \mathsf{Enc}(pk, m_\beta);$
 $g \leftarrow \mathcal{A}(c);$
 return $g = \beta$;

In the above game, we require $m_0, m_1 \in \mathfrak{m}$ *and* $|m_0| = |m_1|$*. We say* Γ *satisfies* IND-CPA*, if for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, there exists a negligible function* negl*, such that for all security parameters* κ*, we have* $\mathsf{Succ}(\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \kappa)) \leq 1/2 + \mathsf{negl}(\kappa)$*.*

Definition 8 (IND-PA0 [4])**.** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an asymmetric encryption scheme,* $\mathcal{A}$ *be an adversary,* κ *be the security parameter, and* $\mathsf{IND\text{-}PA0}(\Pi, \mathcal{A}, \kappa)$ *be the following game.*

$\mathsf{IND\text{-}PA0}(\Pi, \mathcal{A}, \kappa) =$

---

[6] We shall implicitly bind ternary operators occasionally, i.e., we write Γ *is a homomorphic asymmetric encryption scheme* as opposed to the more verbose Γ *is a homomorphic asymmetric encryption scheme, with respect to ternary operators* $\odot$*,* $\oplus$*, and* $\otimes$*.*

[7] We write $X \circ_{pk} Y$ for the application of ternary operator $\circ$ to inputs X, Y, and pk. We occasionally abbreviate $X \circ_{pk} Y$ as $X \circ Y$, when pk is clear from the context.

[8] Our definition of an asymmetric encryption scheme explicitly defines the plaintext space, whereas, Bellare *et al.* [3] leave the plaintext space implicit; this change is reflected in our definition of IND-CPA. Moreover, we provide the adversary with the message space and security parameter. We adapt IND-PA0 similarly.

$(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(\kappa)$;
$(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathfrak{m}, \kappa)$;
$\beta \leftarrow_R \{0,1\}$;
$c \leftarrow \mathsf{Enc}(pk, m_\beta)$;
$\mathbf{c} \leftarrow \mathcal{A}(c)$;
$\mathbf{m} \leftarrow (\mathsf{Dec}(sk, \mathbf{c}[1]), \ldots, \mathsf{Dec}(sk, \mathbf{c}[|\mathbf{c}|]))$;
$g \leftarrow \mathcal{A}(\mathbf{m})$;
return $g = \beta \wedge \bigwedge_{1 \leq i \leq |\mathbf{c}|} c \neq \mathbf{c}[i]$;

In the above game, we require $m_0, m_1 \in \mathfrak{m}$ and $|m_0| = |m_1|$. We say Γ satisfies IND-PA0*, if for all probabilistic polynomial-time adversaries $\mathcal{A}$, there exists a negligible function* negl*, such that for all security parameters κ, we have* $\mathsf{Succ}(\mathsf{IND\text{-}PA0}(\Pi, \mathcal{A}, \kappa)) \leq 1/2 + \mathsf{negl}(\kappa)$.

A.2 Proof systems

Definition 9 (from [68])**.** *Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a homomorphic asymmetric encryption scheme and Σ be a sigma protocol for a binary relation R.*[9]

- Σ proves correct key generation *if a* $((\kappa, pk, \mathfrak{m}), (sk, s)) \in R \Leftrightarrow (pk, sk, \mathfrak{m}) = \mathsf{Gen}(\kappa; s)$.

Further, suppose that $(pk, sk, \mathfrak{m})$ is the output of $\mathsf{Gen}(\kappa; s)$, for some security parameter κ and coins s.

- Σ proves plaintext knowledge in a subspace *if* $((pk, c, \mathfrak{m}'), (m, r)) \in R \Leftrightarrow c = \mathsf{Enc}(pk, m; r) \wedge m \in \mathfrak{m}' \wedge \mathfrak{m}' \subseteq \mathfrak{m}$. *We call $\mathfrak{m}'$ the subspace.*

- Σ proves correct decryption *if* $((pk, c, m), sk) \in R \Leftrightarrow m = \mathsf{Dec}(sk, c)$.

Definition 10 (Non-interactive proof system [68])**.** *A* non-interactive proof system *for a relation R is a tuple of algorithms $(\mathsf{Prove}, \mathsf{Verify})$, such that:*

- **Prove**, *denoted $\sigma \leftarrow \mathsf{Prove}(s, w, \kappa)$, is executed by a prover to prove $(s, w) \in R$.*

- **Verify**, *denoted $v \leftarrow \mathsf{Verify}(s, \sigma, \kappa)$, is executed by anyone to check the validity of a proof. We assume* Verify *is deterministic.*

Moreover, the system must be complete*: there exists a negligible function* negl*, such that for all statement and witnesses $(s, w) \in R$ and security parameters κ, we have* $\Pr[\sigma \leftarrow \mathsf{Prove}(s, w, \kappa) : \mathsf{Verify}(s, \sigma, \kappa) = 1] > 1 - \mathsf{negl}(\kappa)$.

Definition 11 (Fiat-Shamir transformation [33])**.** *Given a sigma protocol* $\Sigma = (\mathsf{Comm}, \mathsf{Chal}, \mathsf{Resp}, \mathsf{Verify}_\Sigma)$ *for relation R and a hash function $\mathcal{H}$, the* Fiat-Shamir transformation*, denoted $\mathsf{FS}(\Sigma, \mathcal{H})$, is the tuple $(\mathsf{Prove}, \mathsf{Verify})$ of algorithms, defined as follows:*

[9] Given a binary relation R, we write $((s_1, \ldots, s_l), (w_1, \ldots, w_k)) \in R \Leftrightarrow P(s_1, \ldots, s_l, w_1, \ldots, w_k)$ for $(s, w) \in R \Leftrightarrow P(s_1, \ldots, s_l, w_1, \ldots, w_k) \wedge s = (s_1, \ldots, s_l) \wedge w = (w_1, \ldots, w_k)$, hence, R is only defined over pairs of vectors of lengths l and k.

$\mathsf{Prove}(s, w, \kappa) =$

$(\mathsf{comm}, t) \leftarrow \mathsf{Comm}(s, w, \kappa);$
$\mathsf{chal} \leftarrow \mathcal{H}(\mathsf{comm}, s);$
$\mathsf{resp} \leftarrow \mathsf{Resp}(\mathsf{chal}, t, \kappa);$
$\textbf{return } (\mathsf{comm}, \mathsf{resp});$

$\mathsf{Verify}(s, (\mathsf{comm}, \mathsf{resp}), \kappa) =$

$\mathsf{chal} \leftarrow \mathcal{H}(\mathsf{comm}, s);$
$\textbf{return } \mathsf{Verify}_\Sigma(s, (\mathsf{comm}, \mathsf{chal}, \mathsf{resp}), \kappa);$

A string m can be included in the hashes computed by algorithms Prove *and* Verify*. That is, the hashes are computed in both algorithms as* $\mathsf{chal} \leftarrow \mathcal{H}(\mathsf{comm}, s, m)$*. We write* $\mathsf{Prove}(s, w, m, \kappa)$ *and* $\mathsf{Verify}(s, (\mathsf{comm}, \mathsf{resp}), m, k)$ *for invocations of* Prove *and* Verify *which include string m.*

Definition 12 (Zero-knowledge [57])**.** *Let $\Delta = (\mathsf{Prove}, \mathsf{Verify})$ be a non-interactive proof system for a relation R, derived by application of the Fiat-Shamir transformation [33] to a random oracle $\mathcal{H}$ and a sigma protocol. Moreover, let $\mathcal{S}$ be an algorithm, $\mathcal{A}$ be an adversary, κ be a security parameter, and $ZK(\Delta, \mathcal{A}, \mathcal{H}, \mathcal{S}, \kappa)$ be the following game.*

$\mathsf{ZK}(\Delta, \mathcal{A}, \mathcal{H}, \mathcal{S}, \kappa) =$

$\beta \leftarrow_R \{0, 1\};$
$g \leftarrow \mathcal{A}^{\mathcal{H}, \mathcal{P}}(\kappa);$
$\textbf{return } g = \beta;$

Oracle $\mathcal{P}$ is defined on inputs $(s, w) \in R$ as follows:

- *$\mathcal{P}(s, w)$ computes* **if** $\beta = 0$ **then** $\sigma \leftarrow \mathsf{Prove}(s, w, \kappa)$ **else** $\sigma \leftarrow \mathcal{S}(s, \kappa)$ *and outputs σ.*

And algorithm $\mathcal{S}$ can patch random oracle $\mathcal{H}$.[10] *We say Δ satisfies* zero-knowledge*, if there exists a probabilistic polynomial-time algorithm $\mathcal{S}$, such that for all probabilistic polynomial-time algorithm adversaries $\mathcal{A}$, there exists a negligible function* negl*, and for all security parameters κ, we have* $\mathsf{Succ}(ZK(\Delta, \mathcal{A}, \mathcal{H}, \mathcal{S}, \kappa)) \leq \frac{1}{2} + \mathsf{negl}(\kappa)$*. An algorithm $\mathcal{S}$ for which zero-knowledge holds is called a* simulator *for* (Prove, Verify)*.*

Definition 13 (Simulation sound extractability [68, 10, 39])**.** *Suppose Σ is a sigma protocol for relation R, $\mathcal{H}$ is a random oracle, and* (Prove, Verify) *is a non-interactive proof system, such that* $\mathsf{FS}(\Sigma, \mathcal{H}) = (\mathsf{Prove}, \mathsf{Verify})$*. Further suppose $\mathcal{S}$ is a simulator for* (Prove, Verify) *and $\mathcal{H}$ can be patched by $\mathcal{S}$. Proof system* (Prove, Verify) *satisfies* simulation sound extractability *if there exists a probabilistic polynomial-time algorithm $\mathcal{K}$, such that for all probabilistic polynomial-time adversaries $\mathcal{A}$ and coins r, there exists a negligible function* negl*, such that*

[10] Random oracles can be *programmed* or *patched*. We will not need the details of how patching works, so we omit them here; see Bernhard et al. [10] for a formalisation.

for all security parameters κ, we have:[11]

$$\begin{aligned}\Pr[&\mathbf{P} \leftarrow (); \mathbf{Q} \leftarrow \mathcal{A}^{\mathcal{H},\mathcal{P}}(\text{—}; r); \mathbf{W} \leftarrow \mathcal{K}^{\mathcal{A}'}(\mathbf{H}, \mathbf{P}, \mathbf{Q}) : \\ &\quad |\mathbf{Q}| \neq |\mathbf{W}| \vee \exists j \in \{1, \ldots, |\mathbf{Q}|\} \,.\, (\mathbf{Q}[j][1], \mathbf{W}[j]) \notin R \wedge \\ &\qquad \forall (s, \sigma) \in \mathbf{Q}, (t, \tau) \in \mathbf{P} \,.\, \mathsf{Verify}(s, \sigma, \kappa) = 1 \wedge \sigma \neq \tau] \leq \mathsf{negl}(\kappa)\end{aligned}$$

where $\mathcal{A}(\text{—}; r)$ denotes running adversary $\mathcal{A}$ with an empty input and coins r, where $\mathbf{H}$ is a transcript of the random oracle's input and output, and where oracles $\mathcal{A}'$ and $\mathcal{P}$ are defined below:

- $\mathcal{A}'()$. *Computes $\mathbf{Q}' \leftarrow \mathcal{A}(\text{—}; r)$, forwarding any of $\mathcal{A}$'s oracle queries to $\mathcal{K}$, and outputs $\mathbf{Q}'$. By running $\mathcal{A}(\text{—}; r)$, $\mathcal{K}$ is rewinding the adversary.*
- $\mathcal{P}(s)$. *Computes $\sigma \leftarrow \mathcal{S}(s); \mathbf{P} \leftarrow (\mathbf{P}[1], \ldots, \mathbf{P}[|\mathbf{P}|], (s, \sigma))$ and outputs σ.*

Algorithm $\mathcal{K}$ is an extractor *for* $(\mathsf{Prove}, \mathsf{Verify})$.

Theorem 5 (from [10])**.** *Let Σ be a sigma protocol for relation R, and let $\mathcal{H}$ be a random oracle. Suppose Σ satisfies special soundness and special honest verifier zero-knowledge. Non-interactive proof system $\mathsf{FS}(\Sigma, \mathcal{H})$ satisfies zero-knowledge and simulation sound extractability.*

B Helios

Smyth, Frink & Clarkson [68] formalise a generic construction for Helios-like election schemes (Definition 14), which can be instantiated to derive Helios'16 (Definition 15).

Definition 14 (Generalised Helios [68])**.** *Suppose* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is an additively homomorphic asymmetric encryption scheme, Σ_1 is a sigma protocol that proves correct key generation, Σ_2 is a sigma protocol that proves plaintext knowledge in a subspace, Σ_3 is a sigma protocol that proves correct decryption, and $\mathcal{H}$ is a hash function. Let* $\mathsf{FS}(\Sigma_1, \mathcal{H}) = (\mathsf{ProveKey}, \mathsf{VerKey})$, $\mathsf{FS}(\Sigma_2, \mathcal{H}) = (\mathsf{ProveCiph}, \mathsf{VerCiph})$, *and* $\mathsf{FS}(\Sigma_3, \mathcal{H}) = (\mathsf{ProveDec}, \mathsf{VerDec})$. *We define election scheme* generalised Helios*, denoted* $\mathsf{Helios}(\Pi, \Sigma_1, \Sigma_2, \Sigma_3, \mathcal{H}) = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$*, as follows.*

- $\mathsf{Setup}(\kappa)$. *Select coins s uniformly at random, compute* $(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(\kappa; s); \rho \leftarrow \mathsf{ProveKey}((\kappa, pk, \mathfrak{m}), (sk, s), \kappa); pk' \leftarrow (pk, \mathfrak{m}, \rho); sk' \leftarrow (pk, sk)$*, let m be the largest integer such that $\{0, \ldots, m\} \subseteq \{0\} \cup \mathfrak{m}$, and output (pk', sk', m, m).*
- $\mathsf{Vote}(pk', v, nc, \kappa)$. *Parse pk' as a vector $(pk, \mathfrak{m}, \rho)$. Output $\perp$ if parsing fails or* $\mathsf{VerKey}((\kappa, pk, \mathfrak{m}), \rho, \kappa) \neq 1 \vee v \notin \{1, \ldots, nc\}$. *Select coins $r_1, \ldots, r_{nc-1}$ uniformly at random and compute:*

[11] We extend set membership notation to vectors: we write $x \in \mathbf{x}$ if x is an element of the set $\{\mathbf{x}[i] : 1 \leq i \leq |\mathbf{x}|\}$.

> **for** $1 \le j \le nc-1$ **do**
>   **if** $j = v$ **then** $m_j \leftarrow 1$; **else** $m_j \leftarrow 0$;
> $c_j \leftarrow \mathsf{Enc}(pk, m_j; r_j)$;
> $\sigma_j \leftarrow \mathsf{ProveCiph}((pk, c_j, \{0,1\}), (m_j, r_j), j, \kappa)$;
> $c \leftarrow c_1 \otimes \cdots \otimes c_{nc-1}$;
> $m \leftarrow m_1 \odot \cdots \odot m_{nc-1}$;
> $r \leftarrow r_1 \oplus \cdots \oplus r_{nc-1}$;
> $\sigma_{nc} \leftarrow \mathsf{ProveCiph}((pk, c, \{0,1\}), (m, r), nc, \kappa)$;

Output ballot $(c_1, \ldots, c_{nc-1}, \sigma_1, \ldots, \sigma_{nc})$.

- $\mathsf{Tally}(sk', \mathfrak{bb}, nc, \kappa)$. *Initialise vectors* $\mathfrak{v}$ *of length* nc *and* pf *of length* $nc-1$. *Compute* **for** $1 \le j \le nc$ **do** $\mathfrak{v}[j] \leftarrow 0$. *Parse* sk' *as a vector* (pk, sk). *Output* $(\mathfrak{v}, pf)$ *if parsing fails. Let* $\{b_1, \ldots, b_\ell\}$ *be the largest subset of* $\mathfrak{bb}$ *such that* $b_1 < \cdots < b_\ell$ *and for all* $1 \le i \le \ell$ *we have* b_i *is a vector of length* $2 \cdot nc - 1$ *and* $\bigwedge_{j=1}^{nc-1} \mathsf{VerCiph}((pk, b_i[j], \{0,1\}), b_i[j+nc-1], j, \kappa) = 1 \wedge \mathsf{VerCiph}((pk, b_i[1] \otimes \cdots \otimes b_i[nc-1], \{0,1\}), b_i[2 \cdot nc - 1], nc, \kappa) = 1$. *If* $\{b_1, \ldots, b_\ell\} = \emptyset$, *then output* $(\mathfrak{v}, pf)$, *otherwise, compute:*

  > **for** $1 \le j \le nc-1$ **do**
  > $c \leftarrow b_1[j] \otimes \cdots \otimes b_\ell[j]$;
  > $\mathfrak{v}[j] \leftarrow \mathsf{Dec}(sk, c)$;
  > $pf[j] \leftarrow \mathsf{ProveDec}((pk, c, \mathfrak{v}[j]), sk, k)$;
  > $\mathfrak{v}[nc] \leftarrow \ell - \sum_{j=1}^{nc-1} \mathfrak{v}[j]$;

  Output $(\mathfrak{v}, pf)$.

- $\mathsf{Verify}(pk', \mathfrak{bb}, nc, \mathfrak{v}, pf, \kappa)$. *Parse* $\mathfrak{v}$ *as a vector of length* nc, *parse* pf *as a vector of length* $nc-1$, *parse* pk' *as a vector* $(pk, \mathfrak{m}, \rho)$. *Output* 0 *if parsing fails or* $\mathsf{VerKey}((\kappa, pk, \mathfrak{m}), \rho, \kappa) \neq 1$. *Let* $\{b_1, \ldots, b_\ell\}$ *be the largest subset of* $\mathfrak{bb}$ *satisfying the conditions given by the tally algorithm and let* mb *be the largest integer such that* $\{0, \ldots, mb\} \subseteq \mathfrak{m}$. *If* $\{b_1, \ldots, b_\ell\} = \emptyset \wedge \bigwedge_{j=1}^{nc} \mathfrak{v}[j] = 0$ *or* $\bigwedge_{j=1}^{nc-1} \mathsf{VerDec}((pk, b_1[j] \otimes \cdots \otimes b_\ell[j], \mathfrak{v}[j]), pf[j], k) = 1 \wedge \mathfrak{v}[nc] = \ell - \sum_{j=1}^{nc-1} \mathfrak{v}[j] \wedge 1 \le \ell \le mb$, *then output* 1, *otherwise, output* 0.

The above algorithms assume $nc > 1$. *Smyth, Frink & Clarkson define special cases of* Vote, Tally *and* Verify *when* $nc = 1$. *We omit those cases for brevity and, henceforth, assume* nc *is always greater than one.*

Definition 15 (Helios'16 [68]). *Election scheme* Helios'16 *is* $\mathsf{Helios}(\Pi, \Sigma_1, \Sigma_2, \Sigma_3, \mathcal{H})$, *where* Π *is additively homomorphic El Gamal [24, §2],* Σ_1 *is the sigma protocol for proving knowledge of discrete logarithms by Chaum et al. [16, Protocol 2],* Σ_2 *is the sigma protocol for proving knowledge of disjunctive equality between discrete logarithms by Cramer et al. [23, Figure 1],* Σ_3 *is the sigma protocol for proving knowledge of equality between discrete logarithms by Chaum & Pedersen [17, §3.2], and* $\mathcal{H}$ *is a random oracle.*

Although Helios actually uses SHA-256 [52], we assume that $\mathcal{H}$ is a random oracle to prove Theorem 4. Moreover, we assume the sigma protocols used by Helios'16 satisfy the preconditions of generalised Helios, that is, [16, Protocol 2] is a sigma protocol for proving correct key generation, [23, Figure 1] is a sigma protocol for proving plaintext knowledge in a subspace, and [17, §3.2] is a sigma protocol for proving decryption. We leave formally proving this assumption as future work.

C Proofs

C.1 Proof of Lemma 1 (correctness)

Let $\Pi = (\mathsf{Gen}_\Pi, \mathsf{Enc}_\Pi, \mathsf{Dec}_\Pi)$, $\Delta = (\mathsf{Prove}, \mathsf{Verify})$, and $\gamma(\Pi, \Delta, \ell) = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Moreover, let $\mathcal{H}$ denote the hash function. Suppose κ is a security parameter and m is a message. Further suppose $(pk, sk, \mathfrak{m})$ is an output of $\mathsf{Gen}(\kappa)$ such that $m \in \mathfrak{m}$ and c is an output of $\mathsf{Enc}(pk, m)$. By definition of the Gen, we have $\mathfrak{m} = \{(m_1, \ldots, m_\ell) \mid m_1, \ldots, m_\ell \in \mathfrak{M} \wedge m_1 \odot \cdots \odot m_\ell \in \mathfrak{M}\}$, where $\mathfrak{M}$ is the subspace. Hence, m is a block of messages $(m_1, \ldots, m_\ell)$. By definition of Enc, ciphertext c is a tuple $(c_1, \sigma_1, \ldots, c_\ell, \sigma_\ell, \sigma)$ such that $c_i = \mathsf{Enc}_\Pi(pk, m_i; r_i)$ and σ_i is an output of $\mathsf{Prove}((pk, c_i, \mathfrak{M}), (m_i, r_i), \kappa)$, where $1 \leq i \leq \ell$ and coins r_i are chosen uniformly at random. Moreover, σ is an output of $\mathsf{Prove}((pk, c, \mathfrak{M}), (m, r), \kappa)$, where $c = c_1 \otimes \cdots \otimes c_\ell$, $m = m_1 \odot \cdots \odot m_\ell$, and $r = r_1 \oplus \cdots \oplus r_\ell$. By completeness of Δ, we have $\mathsf{Verify}((pk, c, \mathfrak{M}), \sigma, \kappa) \wedge \bigwedge_{1 \leq i \leq \ell} \mathsf{Verify}((pk, c_i, \mathfrak{M}), \sigma_i, \kappa)$, with overwhelming probability. Hence, $\mathsf{Dec}(sk, c)$ outputs $(\mathsf{Dec}_\Pi(sk, c_1), \ldots, \mathsf{Dec}_\Pi(sk, c_\ell))$, with overwhelming probability. And, by correctness of Π, we have $(\mathsf{Dec}_\Pi(sk, c_1), \ldots, \mathsf{Dec}_\Pi(sk, c_\ell)) = m$, with overwhelming probability. Thereby concluding our proof. □

C.2 Proof of Theorem 2 (non-malleability)

Let $\Pi = (\mathsf{Gen}_\Pi, \mathsf{Enc}_\Pi, \mathsf{Dec}_\Pi)$, $\Delta = (\mathsf{Prove}, \mathsf{Verify})$, and $\Gamma = \gamma(\Pi, \Delta, \ell) = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Moreover, let $\mathfrak{M}$ be the subspace used by Δ. Proof system Δ satisfies zero-knowledge and simulation sound extractability, because the underlying sigma protocol satisfies special soundness and special honest verifier zero-knowledge (Theorem 5). Hence, there exists a simulator $\mathcal{S}$ and a extractor $\mathcal{K}$ for Δ. Suppose Γ does not satisfy CNM-CPA. Hence, Γ does not satisfy IND-PA0 either [4], and there exists an adversary that wins IND-PA0 against Γ. We proceed with a sequence of games.[12]

C.2.1 Simulate decryption.

Let G be the game derived from IND-PA0 by replacing $\mathbf{m} \leftarrow (\mathsf{Dec}(sk, \mathbf{c}[1]), \ldots, \mathsf{Dec}(sk, \mathbf{c}[|\mathbf{c}|])$ with $\mathbf{m} \leftarrow D(\mathbf{c})$, where algorithm D exploits extractor $\mathcal{K}$ to simulate decryption without private key sk. Namely,

[12] Shoup presents a brief tutorial on structuring proofs as sequences of games [63].

- $D(\mathbf{c})$ proceeds as follows. Parse $\mathbf{c}$ as vector $(\mathbf{c}_1, \dots, \mathbf{c}_{|\mathbf{c}|})$. Initialises $\mathbf{m}$ as a vector of length $|\mathbf{c}|$ and $\mathbf{Q}$ as an empty vector. For each $i \in \{1, \dots, |\mathbf{c}|\}$ process $\mathbf{c}_i$ as follows: if ciphertext $\mathbf{c}_i$ is a valid ciphertext, i.e., $\mathbf{c}_i$ is a vector of length $2 \cdot \ell + 1$ and all its proofs hold,[13] then compute $\mathbf{Q} \leftarrow (\mathbf{Q}[1], \dots, \mathbf{Q}[|\mathbf{Q}|], ((pk, \mathbf{c}_i[1], \mathfrak{M}), \mathbf{c}_i[2]), \dots, ((pk, \mathbf{c}_i[2 \cdot \ell - 1], \mathfrak{M}), \mathbf{c}_i[2 \cdot \ell]))$, otherwise, compute $\mathbf{m}[i] \leftarrow \perp$. Initialise $\mathbf{H}$ as a transcript of the random oracle's input and output, and $\mathbf{P}$ as a transcript of simulated proofs. Compute

  $\mathbf{W} \leftarrow \mathcal{K}(\mathbf{H}, \mathbf{P}, \mathbf{Q})$;
  $i \leftarrow 1$;
  for $j \in \{1, \dots, |\mathbf{m}|\}$ **do**
      if $\mathbf{m}[j] \neq \perp$ **then**
          $\mathbf{m}[j] \leftarrow (\mathbf{W}[i][1], \dots, \mathbf{W}[i + \ell - 1][1])$;
          $i \leftarrow i + \ell$;

We prove that games G and $\mathsf{IND\text{-}PA0}$ are equivalent.

Game G computes $\mathbf{m}[i] = \perp$ when $\mathbf{c}_i$ is not a valid ciphertext, where $1 \leq i \leq |\mathbf{c}|$. By inspection of algorithm Dec, game $\mathsf{IND\text{-}PA0}$ similarly computes $\mathbf{m}[i] = \perp$. Hence, to determine whether games G and $\mathsf{IND\text{-}PA0}$ are equivalent, it suffices to check computations for valid ciphertexts.

By simulation sound extractability, we have for all $i \in \{1, \dots, |\mathbf{c}|\}$, if $\mathbf{c}_i$ is a valid ciphertext, then for all $j \in \{1, \dots, \ell\}$ there exists a message $m_{i,j}$ and coins $r_{i,j}$ such that $\mathbf{c}_i[2 \cdot j - 1] = \mathsf{Enc}_\Pi(pk, m_{i,j}; r_{i,j})$ and $\mathbf{c}_i[2 \cdot j] = \mathsf{Prove}((pk, \mathbf{c}_i[2 \cdot j - 1], \mathfrak{M}), (m_{i,j}, r_{i,j}), i, \kappa)$, with overwhelming probability. Hence, $\mathbf{Q}$ is a tuple of statements and proofs for ciphertexts embedded in valid ciphertexts, and $\mathbf{W}$ contains the corresponding witnesses, i.e., pairs of messages and coins. It follows that Game G computes $\mathbf{m}[i]$ as the plaintext corresponding to ciphertext $\mathbf{c}_i$, for valid ciphertexts, with overwhelming probability. Moreover, although ciphertexts constructed using Enc_Π may not have been constructed using coins chosen uniformly at random, we nevertheless have that game $\mathsf{IND\text{-}PA0}$ also computes $\mathbf{m}[i]$ as the plaintext corresponding to ciphertext $\mathbf{c}_i$, because Π is perfectly correct. Thus, games G and $\mathsf{IND\text{-}PA0}$ are equivalent. Since there exists an adversary that wins $\mathsf{IND\text{-}PA0}$ against Γ, there must also exist an adversary $\mathcal{A}$ that wins G against Γ, i.e., for all negligible functions negl, there exists a security parameter κ such that $\frac{1}{2} + \mathsf{negl}(\kappa) < \mathsf{Succ}(\mathsf{G}(\Gamma, \mathcal{A}, \kappa))$.

C.2.2 Hybrid games.

Let $\mathsf{G0}$, respectively $\mathsf{G1}$, be the game derived from G by replacing $\beta \leftarrow_R \{0, 1\}$ with $\beta \leftarrow 0$, respectively $\beta \leftarrow 1$. These games are trivially related to G, namely,

$$\mathsf{Succ}(\mathsf{G}(\Gamma, \mathcal{A}, \kappa)) = \frac{1}{2} \cdot \mathsf{Succ}(\mathsf{G0}(\Gamma, \mathcal{A}, \kappa)) + \frac{1}{2} \cdot \mathsf{Succ}(\mathsf{G1}(\Gamma, \mathcal{A}, \kappa))$$

[13] All its proofs hold if $\mathsf{Verify}((pk, \mathbf{c}_i[1] \otimes \mathbf{c}_i[3] \otimes \dots \otimes \mathbf{c}_i[2 \cdot \ell - 1], \mathfrak{M}), \mathbf{c}_i[|\mathbf{c}_i|], \ell + 1, \kappa) \wedge \bigwedge_{1 \leq j \leq \ell} \mathsf{Verify}((pk, \mathbf{c}_i[2 \cdot j - 1], \mathfrak{M}), \mathbf{c}_i[2 \cdot j], j, \kappa)$.

We define hybrid games $H_1, \dots, H_\ell$ such that H_i is derived from $\mathsf{G0}$ by replacing $c \leftarrow \mathsf{Enc}(pk, m_\beta)$ with $c \leftarrow \mathsf{Enc}(pk, (m_1[1], \dots, m_1[i], m_0[i+1], \dots, m_0[\ell]))$. It follows that a ciphertext is computed for plaintext m_0 in both $\mathsf{G0}(\Gamma, \mathcal{A}, \kappa)$ and H_0, hence,

$$\mathsf{Succ}(\mathsf{G0}(\Gamma, \mathcal{A}, \kappa)) = \mathsf{Succ}(H_0(\Gamma, \mathcal{A}, \kappa))$$

Let $\mathsf{G1{:}0}$ be the game derived from $\mathsf{G1}$ by replacing $g = \beta$ with $g = 0$. These games are trivially related, namely, $\mathsf{Succ}(\mathsf{G1}(\Gamma, \mathcal{A}, \kappa)) = 1 - \mathsf{Succ}(\mathsf{G1{:}0}(\Gamma, \mathcal{A}, \kappa))$. Moreover, we have

$$\mathsf{Succ}(\mathsf{G1{:}0}(\Gamma, \mathcal{A}, \kappa)) = \mathsf{Succ}(H_\ell(\Gamma, \mathcal{A}, \kappa))$$

because a ciphertext is computed for plaintext m_1 in both $\mathsf{G1{:}0}$ and H_ℓ. It follows that

$$\begin{aligned}
\mathsf{Succ}(\mathsf{G}(\Gamma, \mathcal{A}, \kappa)) &= \frac{1}{2} \cdot (\mathsf{Succ}(\mathsf{G0}(\Gamma, \mathcal{A}, \kappa)) + \mathsf{Succ}(\mathsf{G1}(\Gamma, \mathcal{A}, \kappa))) \\
&= \frac{1}{2} + \frac{1}{2} \cdot (\mathsf{Succ}(H_0(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(\mathsf{G1{:}0}(\Gamma, \mathcal{A}, \kappa))) \\
&= \frac{1}{2} + \frac{1}{2} \cdot (\mathsf{Succ}(H_0(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(H_\ell(\Gamma, \mathcal{A}, \kappa)))
\end{aligned}$$

which can be rewritten as a telescoping series

$$= \frac{1}{2} + \frac{1}{2} \cdot \sum_{1 \leq j < \ell} \mathsf{Succ}(H_j(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(H_{j+1}(\Gamma, \mathcal{A}, \kappa))$$

Suppose $\mathsf{Succ}(H_\iota(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(H_{\iota+1}(\Gamma, \mathcal{A}, \kappa))$ is the largest term in the series, where $1 \leq \iota < \ell$. Thus,

$$\leq \frac{1}{2} + \frac{1}{2} \cdot \ell \cdot (\mathsf{Succ}(H_\iota(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(H_{\iota+1}(\Gamma, \mathcal{A}, \kappa)))$$

Moreover, since adversary $\mathcal{A}$ wins G against Γ, we have

$$\frac{1}{2} + \frac{1}{\ell} \cdot \mathsf{negl}(\kappa) < \frac{1}{2} + \frac{1}{2} \cdot (\mathsf{Succ}(H_\iota(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(H_{\iota+1}(\Gamma, \mathcal{A}, \kappa))) \tag{1}$$

Seeking a contradiction, we use $\mathcal{A}$ to construct an adversary that wins IND-CPA against Π.

C.2.3 Simulate proofs.

Let $\mathcal{B}$ be an adversary against IND-CPA that simulates $\mathcal{A}$'s challenger by embedding its challenge ciphertext as the ιth ciphertext in the challenge ciphertext it computes for $\mathcal{A}$. Moreover, the adversary exploits simulator $\mathcal{S}$ to simulate the proof corresponding to the ιth ciphertext and to simulate the proof corresponding to the homomorphic combination of ciphertexts:

$\mathcal{B}(pk, \mathfrak{m}, \kappa)$ computes $\mathfrak{m} \leftarrow \{(m_1, \ldots, m_\ell) \mid m_1, \ldots, m_\ell \in \mathfrak{M} \wedge m_1 \odot \cdots \odot m_\ell \in \mathfrak{M}\}$; $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathfrak{m}, \kappa)$, parses m_0 as vector $(m_{0,1}, \ldots, m_{0,\ell})$ and m_1 as vector $(m_{1,1}, \ldots, m_{1,\ell})$, and outputs $(m_{0,\iota}, m_{1,\iota})$.

$\mathcal{B}(c_\iota)$ picks coins $r_1, \ldots, r_{\iota-1}, r_{\iota+1}, \ldots, r_\ell$, computes

```
for j ∈ {1, ..., ι − 1} do
    c_j ← Enc_pk(m_{1,j}; r_j);
    σ_j ← Prove((pk, c_j, 𝔐), (m_{1,j}, r_j), j, κ);
σ_ι ← S((pk, c_ι, 𝔐), ι, κ);
for j ∈ {ι + 1, ..., ℓ} do
    c_j ← Enc_pk(m_{0,j}; r_j);
    σ_j ← Prove((pk, c_j, 𝔐), (m_{0,j}, r_j), j, κ);
c ← c_1 ⊗ ··· ⊗ c_ℓ;
σ ← S((pk, c, 𝔐), ℓ + 1, κ);
𝐜 ← A((c_1, σ_1, ..., c_ℓ, σ_ℓ, σ));
𝐦 ← D(𝐜);
g ← A(𝐦)
```

and outputs g.

We prove that $\mathcal{B}$ wins $\mathsf{IND\text{-}CPA}$ against Π.

Suppose $(pk, sk, \mathfrak{m})$ is an output of $\mathsf{Gen}(\kappa)$ and $(m_{0,\iota}, m_{1,\iota})$ is an output of $\mathcal{B}(pk, \mathfrak{m}, \kappa)$, where $m_0 = (m_{0,1}, \ldots, m_{0,\ell})$ and $m_1 = (m_{1,1}, \ldots, m_{1,\ell})$ are the vectors computed by $\mathcal{A}$. It is trivial to see that $\mathcal{B}(pk, \mathfrak{m}, \kappa)$ simulates the challenger in both H_ι and $H_{\iota+1}$ to $\mathcal{A}$. Further suppose c_ι is an output of $\mathsf{Enc}(pk, m_\beta)$ for some bit β and we run $\mathcal{B}(c_\iota)$. Let $(c_1, \sigma_1, \ldots, c_\ell, \sigma_\ell)$ be the ciphertext computed by $\mathcal{B}$ and input to $\mathcal{A}$. If $\beta = 0$, then $(c_1, \sigma_1, \ldots, c_\ell, \sigma_\ell)$ simulates the ciphertext constructed by $\mathcal{A}$'s challenger in H_ι, otherwise, it simulates the ciphertext in $H_{\iota+1}$, with overwhelming probability. Using a similar argument, the plaintexts computed by $\mathcal{B}$ and input to $\mathcal{A}$ simulate the plaintexts constructed by $\mathcal{A}$'s challenger in H_ι if $\beta = 0$ and in $H_{\iota+1}$ otherwise, with overwhelming probability. Suppose $\mathcal{B}$ outputs g. Hence, either

- $\beta = 0$ and $\mathcal{B}(c_\iota)$ simulates the challenger in H_ι, thus $g = \beta$ with at least the probability that $\mathcal{A}$ wins H_ι; or
- $\beta = 1$ and $\mathcal{B}(c_\iota)$ simulates the challenger in $H_{\iota+1}$, thus, $g \neq \beta$ with at least the probability that $\mathcal{A}$ looses $H_{\iota+1}$ and, since $\mathcal{A}$ wins game G, we have g is a bit, hence, $g = \beta$.

It follows that $\mathsf{Succ}(\mathsf{IND\text{-}CPA}(\Pi, \mathcal{B}, \kappa))$ is at least $\frac{1}{2} \cdot \mathsf{Succ}(H_\iota(\Gamma, \mathcal{A}, \kappa)) + \frac{1}{2} \cdot (1 - \mathsf{Succ}(H_{\iota+1}(\Gamma, \mathcal{A}, \kappa))) = \frac{1}{2} + \frac{1}{2} \cdot (\mathsf{Succ}(H_\iota(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(H_{\iota+1}(\Gamma, \mathcal{A}, \kappa)))$, hence, by (1), we have

$$\frac{1}{2} + \frac{1}{\ell} \cdot \mathsf{negl}(\kappa) < \mathsf{Succ}(\mathsf{IND\text{-}CPA}(\Pi, \mathcal{B}, \kappa)),$$

thereby deriving a contradiction and concluding our proof. □

C.3 Proof of Theorem 4 (ballot secrecy)

Let $\Gamma = \mathsf{Helios}(\Pi, \Sigma_1, \Sigma_2, \Sigma_3, \mathcal{H})$, where Π, Σ_1, Σ_2, Σ_3 and $\mathcal{H}$ satisfy the preconditions of Definition 14 such that Σ_2 uses subspace $\mathfrak{M} = \{0, 1\}$. Suppose Π satisfies IND-CPA and is perfectly correct, and Σ_1 and Σ_2 satisfy special soundness and special honest verifier zero-knowledge. We prove that Γ satisfies Ballot-Secrecy, which suffices for our result. Indeed, Helios'16 defines Π as additively homomorphic El Gamal [24, §2], Σ_1 as the sigma protocol for proving knowledge of discrete logarithms by Chaum *et al.* [16, Protocol 2], and Σ_2 as the sigma protocol for proving knowledge of disjunctive equality between discrete logarithms by Cramer *et al.* [23, Figure 1]. Hence, Π satisfies IND-CPA [**?**, 44] and is perfectly correct, and Σ_1 and Σ_2 both satisfy special soundness and special honest verifier zero-knowledge [10, §4]

Let $\Pi = (\mathsf{Gen}_\Pi, \mathsf{Enc}_\Pi, \mathsf{Dec}_\Pi)$, $\mathsf{FS}(\Sigma_2, \mathcal{H}) = (\mathsf{ProveCiph}, \mathsf{VerCiph})$, and $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$. Moreover, let $\mathcal{S}$ be a simulator for $\mathsf{FS}(\Sigma_1, \mathcal{H})$. Suppose Γ does not satisfy Ballot-Secrecy. By Theorem 3, Γ does not satisfy IND-CVA either. (We have HB-Tally-Soundness by [64, 68].) Hence, there exists an adversary $\mathcal{A}$ that wins IND-CVA against Γ. Seeking a contradiction, we use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that wins IND-PA0 against our construction γ:

$\mathcal{B}(pk, \mathfrak{m}, \kappa)$ overwrites $\mathfrak{m}$ with Π's message space, computes $\rho \leftarrow \mathcal{S}((\kappa, pk, \mathfrak{m}), \kappa); pk' \leftarrow (pk, \mathfrak{m}, \rho); (v_0, v_1, nc) \leftarrow \mathcal{A}(pk, \kappa)$, initialise $\mathbf{m}_0$ and $\mathbf{m}_1$ as zero-filled vector of length $nc - 1$, assign 1 to $\mathbf{m}_0[v_0]$ if $v_0 < nc$ and, similarly, assign 1 to $\mathbf{m}_1[v_1]$ if $v_1 < nc$, and output $(\mathbf{m}_0, \mathbf{m}_1)$.

$\mathcal{B}(\mathbf{c})$ defines function f such that $f(\mathbf{c})$ parses $\mathbf{c}$ as $(c_1, \sigma_1, \dots, c_\ell, \sigma_\ell, \sigma)$ and outputs $(c_1, \dots, c_\ell, \sigma_1, \dots, \sigma_\ell, \sigma)$, computes $\mathbf{b} \leftarrow f(\mathbf{c}); \mathfrak{bb} \leftarrow \mathcal{A}(\mathbf{b})$, derives the largest subset $\{b_1, \dots, b_k\}$ of $\mathfrak{bb}$ satisfying the conditions of algorithm Tally, and outputs $(f^{-1}(b_1), \dots, f^{-1}(b_k))$.

$\mathcal{B}(\mathbf{m})$ parses $\mathbf{m}$ as a vector $(\mathbf{m}_1, \dots, \mathbf{m}_k)$, initialises $\mathfrak{v}$ as a vector of length nc, computes $\mathfrak{v} \leftarrow \Sigma_{i=1}^{k}(\mathbf{m}_i[1], \dots, \mathbf{m}_i[\ell], 1 - \sum_{j=1}^{\ell} \mathbf{m}_i[j]); g \leftarrow \mathcal{A}(\mathfrak{v})$, and outputs g.

We prove that $\mathcal{B}$ that wins IND-PA0 against our construction γ.

Let $\gamma(\Pi, \mathsf{FS}(\Sigma_2, \mathcal{H}), \ell) = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, for some integer ℓ that we will specify later. Moreover, let $\mathfrak{m}$ be Π's message space. Suppose $(pk, sk, \mathfrak{m}')$ is an output of $\mathsf{Gen}(\kappa)$ and ρ is an output of $\mathcal{S}((\kappa, pk, \mathfrak{m}), \kappa)$. Let $pk' = (pk, \mathfrak{m}, \rho)$. Further suppose (v_0, v_1, nc) is an output of $\mathcal{A}(pk, \kappa)$. Since $\mathcal{S}$ is a simulator for $\mathsf{FS}(\Sigma_1, \mathcal{H})$, we have $\mathcal{B}$ simulates the challenger in IND-CVA to $\mathcal{A}$. In particular, pk' is a triple containing a public key and corresponding message space generated by Gen_Π, and a (simulated) proof of correct key generation. Let us now specify that $\ell = nc - 1$. Moreover, let $\beta \in \{0, 1\}$. Suppose $(\mathbf{m}_0, \mathbf{m}_1)$ is an output of $\mathcal{B}(pk, \mathfrak{m}, \kappa)$ and $\mathbf{c}$ is an output of $\mathsf{Enc}(pk, m_\beta)$. Let $\mathbf{b} = f(\mathbf{c})$. Further suppose $\mathfrak{bb}$ is an output of $\mathcal{A}(\mathbf{b})$. Ciphertext $\mathbf{b}$ is indistinguishable from an output of $\mathsf{Vote}(pk, v_\beta, nc, \kappa)$. Indeed, $\mathbf{b}$ is a tuple $(c_1, \dots, c_\ell, \sigma_1, \dots, \sigma_\ell, \sigma)$ such that $c_j = \mathsf{Enc}_\Pi(pk, \mathbf{m}_\beta[j]; r_j)$ and σ_j is an output of $\mathsf{ProveCiph}((pk, c_j, \{0,1\}), (\mathbf{m}_\beta[j], r_j), j, \kappa)$ for some coins r_j chosen uniformly at random, where $1 \leq j \leq \ell$, and σ is an output of $\mathsf{ProveCiph}((pk, c_1 \otimes \cdots \otimes c_\ell, \{0,1\}), (\mathbf{m}_\beta[1] \odot \cdots \odot \mathbf{m}_\beta[\ell], r_1 \oplus \cdots \oplus r_\ell), \ell+1, \kappa)$. Hence, $\mathcal{B}$ simulates the challenger in $\mathsf{IND\text{-}CVA}$ to $\mathcal{A}$. Suppose $\mathcal{B}$ derives subset $\{b_1, \ldots, b_k\}$ from $\mathfrak{bb}$ and outputs $(f^{-1}(b_1), \ldots, f^{-1}(b_k))$, i.e., $\{b_1, \ldots, b_k\}$ is the largest subset of $\mathfrak{bb}$ satisfying the conditions of algorithm Tally. Let $\mathbf{m} = (\mathsf{Dec}(sk, f^{-1}(b_1)), \ldots, \mathsf{Dec}(sk, f^{-1}(b_k)))$. Further suppose g is an output of $\mathcal{B}(\mathbf{m})$. The following claim proves that $\mathcal{B}(\mathbf{m})$ simulates the challenger in $\mathsf{IND\text{-}CVA}$ to $\mathcal{A}$, hence, $g = \beta$, with at least the probability that $\mathcal{A}$ wins $\mathsf{IND\text{-}CVA}$. Thus, $\mathcal{B}$ wins $\mathsf{IND\text{-}PA0}$ against $\gamma(\Pi, \mathsf{FS}(\Sigma_2, \mathcal{H}), \ell)$, deriving a contradiction (with respect to Theorem 2) and concluding our proof.

Claim. *Adversary $\mathcal{B}$'s computation of $\mathfrak{v}$ is equivalent to computing $\mathfrak{v}$ as $\mathfrak{v} \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, nc, \kappa)$.*

Proof of Claim. Computation $\mathfrak{v} \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, nc, \kappa)$ is equivalent to initialising $\mathfrak{v}$ as a zero-filled vector of length nc and computing

for $1 \leq j \leq nc-1$ **do**
  $\mathfrak{v}[j] \leftarrow \mathsf{Dec}_\Pi(sk, b_1[j] \otimes \cdots \otimes b_k[j])$;
$\mathfrak{v}[nc] \leftarrow k - \sum_{j=1}^{nc-1} \mathfrak{v}[j]$;

By simulation sound extractability, there exists a message $m_{i,j} \in \{0,1\}$ and coins $r_{i,j}$ such that $b_i[j] = \mathsf{Enc}_\Pi(pk, m_{i,j}; r_{i,j})$, with overwhelming probability, where $1 \leq i \leq k$ and $1 \leq j \leq nc-1$. Moreover, we have $\mathbf{m}[i] = (m_{i,1}, \ldots, m_{i,nc-1})$ by correctness of Π, where $1 \leq i \leq k$. Since Π is homomorphic, we have $b_1[j] \otimes \cdots \otimes b_k[j]$ is a ciphertext, with overwhelming probability, where $1 \leq j \leq nc-1$. Moreover, although ciphertext $b_1[j] \otimes \cdots \otimes b_k[j]$ may not have been constructed using coins chosen uniformly at random, we nevertheless have $\mathsf{Dec}_\Pi(sk, b_1[j] \otimes \cdots \otimes b_k[j]) = m_{1,j} \odot \cdots \odot m_{k,j}$, because Π is perfectly correct, where $1 \leq j \leq nc-1$. It follows that the above computation is equivalent to

for $1 \leq j \leq nc-1$ **do**
  $\mathfrak{v}[j] \leftarrow \mathbf{m}[1][j] \odot \cdots \odot \mathbf{m}[k][j]$;
$\mathfrak{v}[nc] \leftarrow k - \sum_{j=1}^{nc-1} \mathfrak{v}[j]$;

Let mb be the largest integer such that $\{0, \ldots, mb\} \subseteq \{0\} \cup \mathfrak{m}$. Since $\mathcal{A}$ is a winning adversary, we have $k \leq mb$. Moreover, since $\odot$ is the addition operator in group $(\mathfrak{m}, \odot)$ and $m_{1,j}, \ldots, m_{k,j} \in \{0,1\}$, we have $m_{1,j} \odot \cdots \odot m_{k,j} = \Sigma_{i=1}^{k} m_{i,j}$, where $1 \leq j \leq nc-1$. It follows that the previous computation is equivalent to

$$\mathfrak{v} \leftarrow (\Sigma_{i=1}^{k} \mathbf{m}[i][1], \ldots, \Sigma_{i=1}^{k} \mathbf{m}[i][nc-1], k - \Sigma_{i=1}^{k} \Sigma_{j=1}^{nc-1} \mathbf{m}[i][j])$$

which is equivalent to adversary $\mathcal{B}$'s computation of $\mathfrak{v}$, concluding our proof. □

References

[1] B. Adida, O. Marneffe, O. Pereira, and J. 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.

[2] R. M. Alvarez and T. E. Hall. *Electronic Elections: The Perils and Promises of Digital Democracy*. Princeton University Press, 2010.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. 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.

[4] M. Bellare and A. Sahai. Non-malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In *CRYPTO'99: 19th International Cryptology Conference*, volume 1666 of *LNCS*, pages 519–536. Springer, 1999.

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

[6] J. Benaloh and M. 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.

[7] J. C. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections. In *STOC'94: 26th Theory of computing Symposium*, pages 544–553. ACM Press, 1994.

[8] D. Bernhard, V. Cortier, D. Galindo, O. Pereira, and B. Warinschi. SoK: A comprehensive analysis of game-based ballot privacy definitions. In *S&P'15: 36th Security and Privacy Symposium*, pages 499–516. IEEE Computer Society, 2015.

[9] D. Bernhard, V. Cortier, O. Pereira, B. Smyth, and B. 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.

[10] D. Bernhard, O. Pereira, and B. Warinschi. How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In *ASIACRYPT'12: 18th International Conference on the Theory and Application of Cryptology and Information Security*, volume 7658 of *LNCS*, pages 626–643. Springer, 2012.

[11] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

[12] D. Boneh and J. Katz. Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In *CT-RSA'05: The Cryptographers' Track at the RSA Conference*, volume 3376 of *LNCS*, pages 87–103. Springer, 2005.

[13] D. Bowen. Secretary of State Debra Bowen Moves to Strengthen Voter Confidence in Election Security Following Top-to-Bottom Review of Voting Systems. California Secretary of State, press release DB07:042 `http://www.sos.ca.gov/elections/voting_systems/ttbr/db07_042_ttbr_system_decisions_release.pdf`, August 2007.

[14] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *CCS'05: 12th ACM Conference on Computer and Communications Security*, pages 320–329. ACM Press, 2005.

[15] R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *EUROCRYPT'04: 23rd International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.

[16] D. Chaum, J. Evertse, J. van de Graaf, and R. 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.

[17] D. Chaum and T. P. Pedersen. Wallet Databases with Observers. In *CRYPTO'92: 12th International Cryptology Conference*, volume 740 of *LNCS*, pages 89–105. Springer, 1993.

[18] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84–90, 1981.

[19] B. Chor, S. Goldwasser, S. Micali, and B. 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.

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

[21] V. Cortier and B. 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.

[22] V. Cortier and B. Smyth. Attacking and fixing Helios: An analysis of ballot secrecy. *Journal of Computer Security*, 21(1):89–148, 2013.

[23] R. Cramer, M. K. Franklin, B. Schoenmakers, and M. 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.

[24] R. Cramer, R. Gennaro, and B. 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.

[25] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In *CRYPTO'98: 18th International Cryptology Conference*, volume 1462 of *LNCS*, pages 13–25. Springer, 1998.

[26] R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *EUROCRYPT'02: 21st International Conference on the Theory and Applications of Cryptographic Techniques*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.

[27] I. Damgård and M. 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.

[28] I. Damgård, M. Jurik, and J. B. Nielsen. A Generalization of Paillier's Public-Key System with Applications to Electronic Voting. *International Journal of Information Security*, 9(6):371–385, 2010.

[29] Y. Desmedt, R. Gennaro, K. Kurosawa, and V. Shoup. A New and Improved Paradigm for Hybrid Encryption Secure Against Chosen-Ciphertext Attack. *Journal of Cryptology*, 23(1):91–120, 2010.

[30] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *STOC'91: 23rd Theory of computing Symposium*, pages 542–552. ACM Press, 1991.

[31] D. Dolev, C. Dwork, and M. Naor. Nonmalleable Cryptography. *Journal on Computing*, 30(2):391–437, 2000.

[32] E. Elkind and A. Sahai. A Unified Methodology For Constructing Public-Key Encryption Schemes Secure Against Adaptive Chosen-Ciphertext Attack. Cryptology ePrint Archive, Report 2002/042, 2002.

[33] A. Fiat and A. Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO'86: 6th International Cryptology Conference*, volume 263 of *LNCS*, pages 186–194. Springer, 1987.

[34] E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *PKC'99: Second International Workshop on Practice and Theory in Public Key Cryptography*, volume 1560 of *LNCS*, pages 53–68. Springer, 1999.

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

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

[37] R. Gonggrijp and W.-J. Hengeveld. Studying the Nedap/Groenendaal ES3B Voting Computer: A Computer Security Perspective. In *EVT'07: Electronic Voting Technology Workshop*. USENIX Association, 2007.

[38] J. Groth. Efficient maximal privacy in boardroom voting and anonymous broadcast. In *FC'04: 8th International Conference on Financial Cryptography*, volume 3110 of *LNCS*, pages 90–104. Springer, 2004.

[39] J. Groth. Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In *ASIACRYPT'02: 12th International Conference on the Theory and Application of Cryptology and Information Security*, volume 4284 of *LNCS*, pages 444–459. Springer, 2006.

[40] A. Gumbel. *Steal This Vote: Dirty Elections and the Rotten History of Democracy in America*. Nation Books, 2005.

[41] F. Hao, P. Y. A. Ryan, and P. Zieliński. Anonymous voting by two-round public discussion. *Journal of Information Security*, 4(2):62 – 67, 2010.

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

[43] M. Hirt. Receipt-Free K-out-of-L Voting Based on ElGamal Encryption. In D. Chaum, M. Jakobsson, R. L. Rivest, and P. Y. Ryan, editors, *Towards Trustworthy Elections: New Directions in Electronic Voting*, volume 6000 of *LNCS*, pages 64–82. Springer, 2010.

[44] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2007.

[45] D. Khader, B. Smyth, P. Y. A. Ryan, and F. Hao. A Fair and Robust Voting System by Broadcast. In *EVOTE'12: 5th International Conference on Electronic Voting*, volume 205 of *Lecture Notes in Informatics*, pages 285–299. Gesellschaft für Informatik, 2012.

[46] S. Khazaei and M. Rezaei-Aliabadi. A rigorous security analysis of a decentralized electronic voting protocol in the universal composability framework. *Journal of Information Security and Applications*, 43:99–109, 2018.

[47] A. Kiayias and M. Yung. Self-tallying elections and perfect ballot secrecy. In *PKC'01: 3rd International Workshop on Practice and Theory in Public Key Cryptography*, volume 2274 of *LNCS*, pages 141–158. Springer, 2002.

[48] A. Lijphart and B. Grofman. *Choosing an electoral system: Issues and Alternatives*. Praeger, 1984.

[49] Y. Lindell. A Simpler Construction of CCA2-Secure Public-Key Encryption Under General Assumptions. In *EUROCRYPT'03: 22nd International Conference on the Theory and Applications of Cryptographic Techniques*, volume 2656 of *LNCS*, pages 241–254. Springer, 2003.

[50] M. Michels and P. Horster. Some Remarks on a Receipt-Free and Universally Verifiable Mix-Type Voting Scheme. In *ASIACRYPT'96: International Conference on the Theory and Application of Cryptology and Information Security*, volume 1163 of *LNCS*, pages 125–132. Springer, 1996.

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

[52] NIST. Secure Hash Standard (SHS). FIPS PUB 180-4, Information Technology Laboratory, National Institute of Standards and Technology, Mar. 2012.

[53] Organization for Security and Co-operation in Europe. *Document of the Copenhagen Meeting of the Conference on the Human Dimension of the CSCE*, 1990.

[54] Organization of American States. *American Convention on Human Rights, "Pact of San Jose, Costa Rica"*, 1969.

[55] B. 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.

[56] B. Pfitzmann and A. 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. Springer, 1989.

[57] E. A. Quaglia and B. Smyth. Secret, verifiable auctions from elections. *Theoretical Computer Science*, 730:44–92, 2018.

[58] E. A. Quaglia and B. Smyth. A short introduction to secrecy and verifiability for elections. arXiv, Report 1702.03168, 2018.

[59] T. Saalfeld. On Dogs and Whips: Recorded Votes. In H. Döring, editor, *Parliaments and Majority Rule in Western Europe*, chapter 16. St. Martin's Press, 1995.

[60] A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *FOCS'99: 40th Annual Symposium on Foundations of Computer Science*, pages 543–553. IEEE Computer Society, 1999.

[61] K. Sako and J. Kilian. Receipt-Free Mix-Type Voting Scheme: A practical solution to the implementation of a voting booth. In *EUROCRYPT'95: 12th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 921 of *LNCS*, pages 393–403. Springer, 1995.

[62] B. 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.

[63] V. Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004.

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

[65] B. Smyth. A foundation for secret, verifiable elections. Cryptology ePrint Archive, Report 2018/225, 2018.

[66] B. Smyth and D. Bernhard. Ballot secrecy and ballot independence coincide. In *ESORICS'13: 18th European Symposium on Research in Computer Security*, volume 8134 of *LNCS*, pages 463–480. Springer, 2013.

[67] B. Smyth and D. Bernhard. Ballot secrecy and ballot independence: definitions and relations. Cryptology ePrint Archive, Report 2013/235, 2014.

[68] B. Smyth, S. Frink, and M. R. Clarkson. Election Verifiability: Cryptographic Definitions and an Analysis of Helios and JCJ. Cryptology ePrint Archive, Report 2015/233, 2017.

[69] B. Smyth, Y. Hanatani, and H. Muratani. NM-CPA secure encryption with proofs of plaintext knowledge. In *IWSEC'15: 10th International Workshop on Security*, volume 9241 of *LNCS*. Springer, 2015.

[70] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman. Security Analysis of the Estonian Internet Voting System. In *CCS'14: 21st ACM Conference on Computer and Communications Security*, pages 703–715. ACM Press, 2014.

[71] United Nations. *Universal Declaration of Human Rights*, 1948.

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

[73] D. 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.

[74] D. Wikström. *Verificatum: How to Implement a Stand-alone Verifier for the Verificatum Mix-Net (VMN Version 3.0.2)*, 2016. `http://www.verificatum.com/files/vmnum-3.0.2.pdf`.

[75] S. Wolchok, E. Wustrow, J. A. Halderman, H. K. Prasad, A. Kankipati, S. K. Sakhamuri, V. Yagati, and R. Gonggrijp. Security Analysis of India's Electronic Voting Machines. In *CCS'10: 17th ACM Conference on Computer and Communications Security*, pages 1–14. ACM Press, 2010.

[76] S. Wolchok, E. Wustrow, D. Isabel, and J. A. Halderman. Attacking the Washington, D.C. Internet Voting System. In *FC'12: 16th International Conference on Financial Cryptography and Data Security*, volume 7397 of *LNCS*, pages 114–128. Springer, 2012.