Secrecy and independence for election schemes

Ben Smyth

Mathematical and Algorithmic Sciences Lab,
Huawei Technologies Co. Ltd., France

October 4, 2015

Abstract

We study ballot secrecy and ballot independence for election schemes. First, we propose a definition of ballot secrecy as an indistinguishability game in the computational model of cryptography. Our definition builds upon and strengthens earlier definitions to ensure that ballot secrecy is preserved in the presence of an adversary that controls the bulletin board and communication channel. Secondly, we propose a definition of ballot independence as an adaptation of a non-malleability definition for asymmetric encryption. We also provide a simpler, equivalent definition as an indistinguishability game. Thirdly, we prove that ballot independence is necessary in election schemes satisfying ballot secrecy. Finally, we demonstrate the applicability of our results by analysing Helios. Our analysis identifies a new attack against Helios, which enables an adversary to determine if a voter did not vote for the adversary's chosen candidate. The attack requires the adversary to control the bulletin board or communication channel, thus, it could not have been detected by earlier definitions of ballot secrecy.

Keywords. Elections, Helios, independence, non-malleability, privacy, secrecy, voting.

1 Introduction

An election is a decision-making procedure to choose representatives. Choices should be made freely, and this has led to a movement towards voting as a secret act. This movement is championed by the United Nations [UN48, Article 21], the Organization for Security and Cooperation in Europe [OSC90, Paragraph 7.4], and the Organization of American States [OAS69, Article 23].

Ballot secrecy[1] is a *de facto* standard requirement of voting systems.

[1] *Ballot secrecy* and *privacy* occasionally appear as synonyms in the literature. We favour ballot secrecy because it avoids confusion with other privacy notions, such as receipt-freeness and coercion resistance, for example.

- *Ballot secrecy.* A voter's vote is not revealed to anyone.

Many voting systems – including systems that have been deployed in real-world, large-scale public elections – attempt to satisfy ballot secrecy by placing extensive trust in software and hardware. Unfortunately, many systems are not trustworthy, and are vulnerable to attacks that could compromise ballot secrecy [GH07, Bow07, WWH+10, WWIH12, SFD+14]. Such vulnerabilities can be avoided by formulating ballot secrecy as a rigorous and precise security definition, and proving that systems satisfy this definition. We propose such a definition in the computational model of cryptography. Our definition builds upon and strengthens earlier definitions of ballot secrecy by Bernhard *et al.* [BCP+11, BPW12b, SB13, SB14, BCG+15b] to ensure that ballot secrecy is preserved in the presence of an adversary that controls the bulletin board and the communication channel, whereas definitions by Bernhard *et al.* only consider trusted bulletin boards and channels.

Ballot independence [Gen95, CS13, CGMA85] is seemingly related to ballot secrecy.

- *Ballot independence.* Observing another voter's interaction with the voting system does not allow a voter to cast a meaningfully related vote, i.e., ballots are non-malleable.

Cortier & Smyth [CS13, CS11, SC11] attribute a class of ballot secrecy attacks to the absence of ballot independence. Their attribution caused some debate. In particular, Bulens, Giry & Pereira [BGP11, §3.2] highlight the investigation of systems which allow the submission of related votes, whilst preserving ballot secrecy, as an interesting research problem. And Desmedt & Chaidos [DC12] claim to provide a solution.[2] We facilitate the study of ballot independence by proposing two cryptographic definitions of independence. Our first definition is a straightforward adaptation of a non-malleability definition for asymmetric encryption. And our second definition is a straightforward adaption of an indistinguishability game for asymmetric encryption. The former definition naturally captures ballot independence, but it is complex and proofs of non-malleability are relatively difficult. The latter definition is equivalent, simpler, and proofs of indistinguishability are easier.

We demonstrate relations between our definitions of secrecy and independence. We prove that ballot secrecy implies ballot independence, hence, ballot independence is necessary (assuming ballot secrecy is required). And show that the inverse implication does not hold, hence, ballot secrecy is strictly stronger than ballot independence.

We employ our ballot secrecy definition to analyse Helios [AMPQ09], a web-based voting system that has been deployed in the real-world. The original Helios scheme is vulnerable to attacks against ballot secrecy [CS13, CS11, SC11]. And the current version of Helios is intended to mitigate against those attacks.

[2] Smyth & Bernhard [SB13, §5.1] critique the results by Desmedt & Chaidos [DC12] and argue that their security results do not support their claims.

Bernhard [Ber14] and Bernhard *et al.* [BCG+15a, BCG+15b] prove that variants of the current version satisfy notions of ballot secrecy, assuming the bulletin board and communication channel are secure, despite the use of malleable ballots. Nevertheless, it follows from our results that ballot secrecy is not satisfied when this assumption is dropped. And this leads to the discovery of a new attack against Helios, whereby an adversary can determine if a voter did not vote for the adversary's chosen candidate.

Contribution. This paper contributes to the security of voting systems by: proposing computational definitions of ballot secrecy (§3) and ballot independence (§4); proving that ballot secrecy is strictly stronger than ballot independence (§5); and identifying a new attack against Helios and proposing a fix (§6).

2 Election schemes

We recall syntax for *election schemes*[3] from Smyth, Frink & Clarkson [SFC15].[4]

Definition 1 (Election scheme [SFC15]). *An* election scheme *is a tuple of efficient algorithms* $(\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally})$ *such that:*

Setup*, denoted*[5] $(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa)$*, is run by the tallier*[6]*.* Setup *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* $b \leftarrow \mathsf{Vote}(pk, nc, v, \kappa)$*, is run by voters.* Vote *takes as input a public key* pk*, some number of candidates* nc*, a voter's vote* v*, and a security parameter* κ*. A voter's vote should be selected from a sequence* $1, \ldots, nc$ *of candidates.* Vote *outputs a ballot* b *or error symbol* $\perp$*.*

Tally*, denoted* $(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, nc, \mathfrak{bb}, \kappa)$*, is run by the tallier.* Tally *takes as input a private key* sk*, some number of candidates* nc*, a bulletin board* $\mathfrak{bb}$*, and a security parameter* κ*, where* $\mathfrak{bb}$ *is a set. It outputs an election outcome* $\mathfrak{v}$ *and a non-interactive tallying proof* pf *(i.e., a proof that the outcome is correct). An election outcome is a vector* $\mathfrak{v}$ *of length* nc *such that* $\mathfrak{v}[v]$ *indicates*[7] *the number of votes for candidate* v*.*

[3] Election schemes capture an interesting class of voting systems, which includes Helios (cf. [SFC15]).

[4] We omit algorithm Verify from our syntax, and we omit the condition that election schemes must satisfy notions of completeness and injectivity, because we do not consider verifiability.

[5] Let $A(x_1, \ldots, x_n; r)$ denote the output of probabilistic algorithm A on inputs $x_1, \ldots, x_n$ and random coins r. Let $A(x_1, \ldots, x_n)$ denote $A(x_1, \ldots, x_n; r)$, where r is chosen uniformly at random. And let $\leftarrow$ denote assignment.

[6] Some election schemes (e.g., Helios) permit the tallier's role to be distributed amongst several talliers. For simplicity, we consider only a single tallier in this paper. Generalising syntax and security definitions to multiple talliers is a possible direction for future work.

[7] Let $\mathfrak{v}[v]$ denote component v of vector $\mathfrak{v}$.

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 if* $\mathfrak{v}$ *is a vector of length* nc *whose components are all* 0*, then*

$$\begin{array}{l}
\Pr[(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa);\\
\quad \textbf{for } 1 \leq i \leq nb \textbf{ do}\\
\quad\quad b_i \leftarrow \mathsf{Vote}(pk, nc, v_i, \kappa);\\
\quad\quad \mathfrak{v}[v_i] \leftarrow \mathfrak{v}[v_i] + 1;\\
\quad (\mathfrak{v}', pf) \leftarrow \mathsf{Tally}(sk, nc, \{b_1, \ldots, b_{nb}\}, \kappa) :\\
\quad nb \leq mb \wedge nc \leq mc \Rightarrow \mathfrak{v} = \mathfrak{v}'] > 1 - \mathsf{negl}(\kappa).
\end{array}$$

3 Ballot Secrecy

Our informal definition of ballot secrecy (§1) could be formulated as an indistinguishability game, similar to indistinguishability games for asymmetric encryption (e.g., IND-CPA): we could challenge the adversary to determine whether a ballot is for one of two possible votes. This formalisation is too weak, because election schemes also output the election outcome and a tallying proof, which needs to be incorporated into the game. Unfortunately, it is insufficient to simply grant the adversary access to an oracle that provides an election outcome and tallying proof corresponding to some ballots, because such a game is unsatisfiable, in particular, the adversary can use the oracle to reveal the vote encapsulated inside the challenge ballot. This reveals some limitations in our informal definition of ballot secrecy.

For simplicity, our informal definition of ballot secrecy deliberately omits some side-conditions, which are necessary for satisfiability, in particular, we did not stress that a voter's vote may be revealed in the following scenarios: unanimous election outcomes reveal how everyone voted and, more generally, election outcomes can be coupled with partial knowledge about the distribution of voters' votes to reveal voters' votes. For example, suppose Alice, Bob and Mallory vote in a referendum and the outcome is two "yes" votes and one "no" vote. Mallory can collude with Alice to reveal Bob's vote. Similarly, Mallory can collude with Bob to reveal Alice's vote. Furthermore, Mallory can reveal that Alice and Bob both voted yes, if she voted no. Accordingly, ballot secrecy must concede that election outcomes reveal partial information about voters' votes,[8] hence, we refine our informal definition of ballot secrecy as follows:

> A voter's vote is not revealed to anyone, except when the vote can be deduced from the election outcome and any partial knowledge on the distribution of votes.

[8] We acknowledge that alternative formalisations of election schemes might permit different results. For instance, voting systems which only announce the winning candidate [BY86,HK02, HK04, DK05], rather than the number of votes for each candidate (i.e., the election outcome, in our terminology), could offer stronger notions of ballot secrecy.

This refinement ensures that the aforementioned examples are not violations of ballot secrecy. By comparison, if Mallory votes yes and can reveal the vote of either Alice or Bob without collusion, then she violates ballot secrecy.

3.1 Indistinguishability game

We formalise ballot secrecy as an indistinguishability game between an adversary and a challenger.[9]

Definition 2 ($\mathsf{Ballot\text{-}Secrecy}$)**.** *Let* $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally})$ *be an election scheme,* $\mathcal{A}$ *be an adversary,* κ *be a security parameter, and* $\mathsf{Ballot\text{-}Secrecy}(\Gamma, \mathcal{A}, \kappa)$ *be the following game.*[10]

$\mathsf{Ballot\text{-}Secrecy}(\Gamma, \mathcal{A}, \kappa) =$

$(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa);$
$nc \leftarrow \mathcal{A}(pk, \kappa);$
$\beta \leftarrow_R \{0, 1\};$
$L \leftarrow \emptyset;$
$\mathfrak{bb} \leftarrow \mathcal{A}^{\mathcal{O}}();$
$(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, nc, \mathfrak{bb}, \kappa);$
$g \leftarrow \mathcal{A}(\mathfrak{v}, pf);$
return $g = \beta \wedge balanced(\mathfrak{bb}, nc, L) \wedge 1 \leq nc \leq mc \wedge |\mathfrak{bb}| \leq mb;$

Predicate $balanced(\mathfrak{bb}, nc, L)$ *holds when: for all votes* $v \in \{1, \ldots, nc\}$ *we have* $|\{b \mid b \in \mathfrak{bb} \wedge \exists v_1 \,.\, (b, v, v_1) \in L\}| = |\{b \mid b \in \mathfrak{bb} \wedge \exists v_0 \,.\, (b, v_0, v) \in L\}|$*. And oracle* $\mathcal{O}$ *is defined as follows:*[11]

- $\mathcal{O}(v_0, v_1)$ *computes* $b \leftarrow \mathsf{Vote}(pk, nc, v_\beta, \kappa); L \leftarrow L \cup \{(b, v_0, v_1)\}$ *and outputs* b*, where* $v_0, v_1 \in \{1, ..., nc\}$*.*

We say Γ *satisfies* ballot secrecy ($\mathsf{Ballot\text{-}Secrecy}$) *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, there exists a negligible function* negl*, such that for all security parameters* κ*, we have* $\mathsf{Succ}(\mathsf{Ballot\text{-}Secrecy}(\Gamma, \mathcal{A}, \kappa)) \leq 1/2 + \mathsf{negl}(\kappa)$*.*

The game captures a setting where the tallier generates a key pair using the scheme's Setup algorithm, publishes the public key, and only uses the private key to compute the election outcome and tallying proof.

The adversary has access to a left-right oracle [BDJR97, BR05] which can compute ballots on the adversary's behalf. Ballots can be computed by the left-right oracle in two ways, corresponding to a bit β chosen uniformly at random by the challenger. If $\beta = 0$, then, given a pair of votes v_0, v_1, the oracle computes

[9] Games are algorithms that output booleans. An adversary *wins* a game by causing it to output true ($\top$). We denote an adversary's *success* $\mathsf{Succ}(\mathsf{Exp}(\cdot))$ in a game $\mathsf{Exp}(\cdot)$ as the probability that the adversary wins, that is, $\mathsf{Succ}(\mathsf{Exp}(\cdot)) = \Pr[\mathsf{Exp}(\cdot) = \top]$. Adversaries are assumed to be *stateful*, that is, information persists across invocations of the adversary in a single game, in particular, the adversary can access earlier assignments.

[10] Let $x \leftarrow_R S$ denote assignment to x of an element chosen uniformly at random from set S. And let $|\mathbf{v}|$ denote the length of vector $\mathbf{v}$.

[11] The oracle may access game parameters, e.g., pk.

a ballot for v_0 and outputs the ballot to the adversary. Otherwise ($\beta = 1$), the oracle outputs a ballot for v_1. The adversary constructs a bulletin board, which may include ballots computed by the oracle. Thus, the game captures a setting where the bulletin board is constructed by an adversary that casts ballots on behalf of a subset of voters and controls the distribution of votes cast by the remaining voters.

The challenger tallies the adversary's bulletin board to derive an election outcome and tallying proof. The adversary is given the outcome and proof, and wins by determining if $\beta = 0$ or $\beta = 1$. Intuitively, if the adversary wins, then there exists a strategy to distinguish ballots. On the other hand, if the adversary loses, then the adversary is unable to distinguish between a ballot for vote v_0 and a ballot for vote v_1, therefore, voters' votes cannot be revealed.

Our notion of ballot secrecy is satisfiable by election schemes which reveal the number of votes for each candidate (i.e., the election outcome). Hence, to avoid trivial distinctions, we insist the game is *balanced*: "left" and "right" inputs to the left-right oracle are equivalent, when the corresponding left-right oracle's outputs appear on the bulletin board. For example, suppose the inputs to the left-right oracle are $(v_{1,0}, v_{1,1}), \dots, (v_{n,0}, v_{n,1})$ and the corresponding outputs are $b_1, \dots, b_n$, further suppose the bulletin board is $\{b_1, \dots, b_\ell\}$ such that $\ell \leq n$; that game is balanced if the "left" inputs $v_{1,0}, \dots, v_{\ell,0}$ are a permutation of the "right" inputs $v_{1,1}, \dots, v_{\ell,1}$. The balanced condition prevents trivial distinctions. For instance, an adversary that constructs a bulletin board containing only the ballot output by a left-right oracle query with input $(1, 2)$ cannot win the game, because it is unbalanced. Albeit, that adversary could trivially determine if $\beta = 0$ or $\beta = 1$, given the tally of that bulletin board.

3.2 Non-malleable encryption is sufficient for secrecy

To demonstrate the applicability of our definition, we recall a construction by Quaglia & Smyth [QS15] for election schemes from asymmetric encryption schemes.[12]

Definition 3 ($\mathsf{Enc2Vote}$ [QS15])**.** *Given an asymmetric encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$,[13] *we define* $\mathsf{Enc2Vote}(\Pi)$ *as follows.*

- $\mathsf{Setup}(\kappa)$ *computes* $(pk, sk) \leftarrow \mathsf{Gen}(\kappa)$ *and outputs* $(pk, sk, poly(\kappa), |\mathfrak{m}|)$.
- $\mathsf{Vote}(pk, nc, v, \kappa)$ *computes* $b \leftarrow \mathsf{Enc}(pk, v)$ *and outputs* b*, if* $1 \leq v \leq nc$*, and outputs* $\perp$*, otherwise.*
- $\mathsf{Tally}(sk, nc, \mathfrak{bb}, \kappa)$ *initialises vector* $\mathbf{v}$ *of length* nc*, computes* **for** $b \in \mathfrak{bb}$ **do** $v \leftarrow \mathsf{Dec}(sk, b)$; **if** $1 \leq v \leq nc$ **then** $\mathbf{v}[v] \leftarrow \mathbf{v}[v] + 1$*, and outputs* $(\mathbf{v}, \epsilon)$.

[12] The construction by Quaglia & Smyth builds upon constructions by Bernhard *et al.* [SB14, SB13, BPW12b, BCP+11].

[13] We define asymmetric encryption and an associated security definition (namely, IND-PA0) in Appendix A.

Algorithm Setup *requires poly to be a polynomial function and* $\mathfrak{m} = \{1, \ldots, |\mathfrak{m}|\}$ *to be the encryption scheme's message space. Algorithm* Tally *requires* ϵ *to be a constant symbol.*

Lemma 1. *Suppose* Π *is an asymmetric encryption scheme with perfect correctness. We have* Enc2Vote(Π) *is an election scheme (i.e.,* Enc2Vote(Π) *satisfies correctness).*

The proof of Lemma 1 appears in [QS15, §D.4].[14]

Intuitively, given a non-malleable asymmetric encryption scheme Π, the construction Enc2Vote(Π) derives ballot secrecy from Π until tallying and algorithm Tally maintains ballot secrecy by returning only the number of votes for each candidate. A formal proof of ballot secrecy follows from Quaglia & Smyth, in particular, Quaglia & Smyth show that Enc2Vote(Π) satisfies a stronger notion of ballot secrecy [QS15, Proposition 5 & 15], hence, Enc2Vote(Π) satisfies our notion of ballot secrecy too.

Corollary 2. *Let* Π *be an encryption scheme with perfect correctness. If* Π *satisfies* IND-PA0*, then election scheme* Enc2Vote(Π) *satisfies* Ballot-Secrecy*.*

The reverse implication of Corollary 2 does not hold.

Proposition 3. *There exists an asymmetric encryption scheme* Π *such that election scheme* Enc2Vote(Π) *satisfies* Ballot-Secrecy*, but* Π *does not satisfy* IND-PA0*.*

The proof of Proposition 3 and all further proofs, except where otherwise stated, appear in Appendix B.

4 Ballot independence

Our informal definition of ballot independence (§1) essentially states that an adversary is unable to construct a ballot meaningfully related to a non-adversarial ballot, i.e., ballots are non-malleable. Hence, we formulate ballot independence using non-malleability. The first formalisation of non-malleability is due to Dolev, Dwork & Naor [DDN91, DDN00], in the context of asymmetric encryption. Bellare & Sahai [BS99] build upon their results, and results by Bellare *et al.* [BDPR98], to introduce an alternative non-malleability definition for asymmetric encryption. We formalise non-malleability for election schemes as a straightforward adaptation of that definition.

Our formalisation of non-malleability for election schemes captures an intuitive notion of ballot independence, but the definition is complex and proofs

[14] Quaglia & Smyth only consider asymmetric encryption schemes with perfect correctness, because they require election schemes to satisfy injectivity, and perfect correctness is required to show that Enc2Vote(Π) satisfies injectivity. We adopt the same assumption to capitalise upon their results.

of non-malleability are relatively difficult. Bellare & Sahai [BS99] observe similar complexities of non-malleability for encryption and show that their non-malleability definition for encryption is equivalent to a simpler, indistinguishability game for encryption. In a similar direction, we derive a simpler, equivalent definition of ballot independence as a straightforward adaptation of that indistinguishability game.

4.1 Non-malleability game

We formalise ballot independence as a non-malleability game.

Definition 4 (CNM-CVA)**.** *Let* $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally})$ *be an election scheme,* $\mathcal{A}$ *be an adversary,* κ *be a security parameter, and* $\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa)$ *and* $\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa)$ *be the following games.*[15]

$\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa) =$

$$\begin{array}{l}
(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa);\\
(V, nc) \leftarrow \mathcal{A}(pk, \kappa);\\
v \leftarrow_R V;\\
b \leftarrow \mathsf{Vote}(pk, nc, v, \kappa);\\
(R, \mathfrak{bb}) \leftarrow \mathcal{A}(b);\\
(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, nc, \mathfrak{bb}, \kappa);\\
\textbf{return } R(v, \mathfrak{v}) \wedge b \notin \mathfrak{bb} \wedge\\
V \subseteq \{1, \dots, nc\}\\
\wedge\ 1 \leq nc \leq mc \wedge |\mathfrak{bb}| \leq mb;
\end{array}$$

$\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa) =$

$$\begin{array}{l}
(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa);\\
(V, nc) \leftarrow \mathcal{A}(pk, \kappa);\\
v, v' \leftarrow_R V;\\
b \leftarrow \mathsf{Vote}(pk, nc, v', \kappa);\\
(R, \mathfrak{bb}) \leftarrow \mathcal{A}(b);\\
(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, nc, \mathfrak{bb}, \kappa);\\
\textbf{return } R(v, \mathfrak{v}) \wedge b \notin \mathfrak{bb}\\
\wedge\ V \subseteq \{1, \dots, nc\}\\
\wedge\ 1 \leq nc \leq mc \wedge |\mathfrak{bb}| \leq mb;
\end{array}$$

In the above games, we insist that relation R *is computable in polynomial time. We say* Γ *satisfies* comparison based non-malleability under chosen vote attack (CNM-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* $\mathsf{Succ}(\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa)) \leq \mathsf{negl}(\kappa)$*.*

Similarly to game Ballot-Secrecy, games cnm-cva and cnm-cva-\$ capture: key generation using algorithm Setup, publication of the public key, and only using the private key to compute the election outcome and tallying proof.

CNM-CVA is satisfied if no adversary can distinguish between games cnm-cva and cnm-cva-\$. That is, for all adversaries, we have with negligible probability that the adversary wins cnm-cva iff the adversary loses cnm-cva-\$. The first three steps of games cnm-cva and cnm-cva-\$ are identical, thus, these steps cannot be distinguished. Then, game cnm-cva-\$ performs an additional step: the challenger samples a second vote v' from vote space V. Thereafter, game $\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa)$, respectively game $\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa)$, proceeds as follows: the challenger constructs a challenge ballot b for v, respectively v'; the adversary is given ballot b and must compute a relation R and bulletin board $\mathfrak{bb}$; the challenger tallies $\mathfrak{bb}$ and outputs the election outcome $\mathfrak{v}$; and the challenger

[15] We abbreviate $x \leftarrow_R S; x' \leftarrow_R S$ as $x, x' \leftarrow_R S$.

evaluates whether $R(v, \mathfrak{v})$ holds. Hence, CNM-CVA is satisfied if there is no advantage of the relation constructed by an adversary given a challenge ballot for v, over the relation constructed by an adversary given a challenge ballot for v'. That is, for all adversaries, we have with negligible probability that the relation evaluated by the challenger in cnm-cva holds iff the relation evaluated in cnm-cva-\$ does not hold. It follows that no adversary can meaningfully relate ballots. On the other hand, if CNM-CVA is not satisfied, then there exists a strategy to construct related ballots.

CNM-CVA avoids crediting the adversary for trivial and unavoidable relations which hold if the challenge ballot appears on the bulletin board. For example, suppose the adversary is given a challenge ballot for v, respectively v', in cnm-cva, respectively cnm-cva-\$, this adversary could output a bulletin board containing only the challenge ballot and a relation R such that $R(v, \mathfrak{v})$ holds if $\mathfrak{v}[v] = 1$, hence, the relation evaluated in cnm-cva holds, whereas the relation evaluated in cnm-cva-\$ does not hold, but the adversary loses in both games because the challenge ballot appears on the bulletin board. By contrast, if the adversary can derive a ballot meaningfully related to the challenge ballot, then the adversary can win the game. For instance, Cortier & Smyth [CS13, CS11] demonstrate the following attack: an adversary observes a voter's ballot, casts a meaningfully related ballot, and exploits the relation to recover the voter's vote from the election outcome.

Comparing CNM-CVA and CNM-CPA. The main distinction between non-malleability for asymmetric encryption (CNM-CPA) and non-malleability for election schemes (CNM-CVA) is: CNM-CPA performs a parallel decryption, whereas, CNM-CVA performs a single tally. It follows that non-malleability for encryption reveals plaintexts corresponding to ciphertexts, whereas, non-malleability for elections reveals the number of ballots for each candidate.

4.2 Indistinguishability game

We formalise an alternative definition of ballot independence as an indistinguishability game.

Definition 5 (IND-CVA)**.** *Let* $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally})$ *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.*

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

$(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa);$
$(v_0, v_1, nc) \leftarrow \mathcal{A}(pk, \kappa);$
$\beta \leftarrow_R \{0, 1\};$
$b \leftarrow \mathsf{Vote}(pk, nc, v_\beta, \kappa);$
$\mathfrak{bb} \leftarrow \mathcal{A}(b);$
$(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, nc, \mathfrak{bb}, \kappa);$
$g \leftarrow \mathcal{A}(\mathfrak{v});$
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* $\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{A}, \kappa) \leq 1/2 + \mathsf{negl}(\kappa)$.

IND-CVA is satisfied if the adversary cannot determine whether the challenge ballot b is for one of two possible votes v_0 and v_1. In addition to the challenge ballot, the adversary is given the election outcome derived by tallying a bulletin board constructed by the adversary. To avoid trivial distinctions, the adversary's bulletin board should not contain the challenge ballot. Intuitively, the adversary wins if there exists a strategy to construct related ballots, since this strategy enables the adversary to construct a ballot b', related to the challenge ballot b, and determine if b is for v_0 or v_1 from the outcome derived by tallying a bulletin board containing b'.

Comparing IND-CVA and IND-PA0. Unsurprisingly, the distinction between indistinguishability for asymmetric encryption (IND-PA0) and indistinguishability for election schemes (IND-CVA), is similar to the distinction between non-malleability for asymmetric encryption and non-malleability for election schemes (§4.1), namely, IND-PA0 performs a parallel decryption, whereas, IND-CVA performs a single tally.

4.3 Equivalence between games

Our ballot independence games are adaptations of standard security definitions for asymmetric encryption: CNM-CVA is based on non-malleability for encryption and IND-CVA is based on indistinguishability for encryption. Bellare & Sahai [BS99] have shown that non-malleability is equivalent to indistinguishability for encryption and their proof can be adapted to show that CNM-CVA and IND-CVA are equivalent.

Theorem 4 (CNM-CVA = IND-CVA)**.** *Given an election scheme* Γ*, we have* Γ *satisfies* CNM-CVA *iff* Γ *satisfies* IND-CVA.

4.4 Non-malleable encryption is sufficient for independence

It follows naturally from our definitions that non-malleable ciphertexts are sufficient for ballot independence. Indeed, we can derive non-malleable ballots in our construction Enc2Vote using encryption schemes satisfying CNM-CPA.[16]

Corollary 5. *Let* Π *be an encryption scheme with perfect correctness. If* Π *satisfies* CNM-CPA*, then election scheme* Enc2Vote(Π) *satisfies* CNM-CVA.

The proof of Corollary 5 follows from Corollary 2 and Theorems 4 & 7. The reverse implication of Corollary 5 does not hold.

[16]Bellare & Sahai [BS99, §5] show that IND-PA0 coincides with CNM-CPA, thus it suffices to consider IND-PA0 in Corollaries 5 & 6.

Corollary 6. *There exists an asymmetric encryption scheme* Π *such that election scheme* $\mathsf{Enc2Vote}(\Pi)$ *satisfies* $\mathsf{CNM\text{-}CVA}$*, but* Π *does not satisfy* $\mathsf{CNM\text{-}CPA}$.

The proof of Corollary 6 follows from Proposition 3 and Theorems 4 & 7.

5 Relations between secrecy and independence

The main distinctions between our ballot secrecy (Ballot-Secrecy) and ballot independence (IND-CVA) games are as follows.

1. The challenger produces one challenge ballot for the adversary in our ballot independence game, whereas, the left-right oracle produces arbitrarily many challenge ballots for the adversary in our ballot secrecy game.
2. The adversary in our ballot secrecy game has access to a tallying proof, but the adversary in our ballot independence game does not.
3. The winning condition in our ballot secrecy game requires the bulletin board to be balanced, whereas, the bulletin board must not contain the challenge ballot in our ballot independence game.

The second point distinguishes our two games and shows that ballot secrecy is stronger than ballot independence.[17] Hence, non-malleable ballots are necessary in election schemes satisfying ballot secrecy.

Theorem 7 (Ballot-Secrecy $\Rightarrow$ IND-CVA)**.** *Let* Γ *be an election scheme. If* Γ *satisfies* Ballot-Secrecy*, then* Γ *satisfies* IND-CVA.

Moreover, since tallying proofs can reveal voters' votes (e.g., a variant of Enc2Vote could define tallying proofs that map ballots to votes) and these proofs are available to the adversary in our ballot secrecy game, but not in our ballot independence game, it follows that ballot secrecy is strictly stronger than ballot independence.

Proposition 8 (IND-CVA $\not\Rightarrow$ Ballot-Secrecy)**.** *There exists an election scheme* Γ *such that* Γ *satisfies* IND-CVA*, but not* Ballot-Secrecy.

The proof of Proposition 8 follows immediately from our informal reasoning and we omit a formal proof.

6 Case Study: Helios

Helios is an open-source, web-based electronic voting system,[18] which has been deployed in the real-world: the International Association of Cryptologic Research (IACR) has used Helios annually since 2010 to elect board members

[17] Smyth & Bernhard explain that alternative formalisations of election schemes might permit different results [SB13, §5.2].

[18] `https://vote.heliosvoting.org`, accessed 19 Aug 2015.

[BVQ10, HBH10],[19] the Catholic University of Louvain used Helios to elect their university president in 2009 [AMPQ09], and Princeton University has used Helios since 2009 to elect student governments [Adi09].[20]

Informally, Helios can be modelled as an election scheme $(\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally})$ such that:

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

Vote enciphers the vote to a ciphertext, proves correct ciphertext construction in zero-knowledge, and outputs the ciphertext coupled with the proof.

Tally proceeds as follows. First, any ballots on the bulletin board for which proofs do not hold are discarded. Secondly, the ciphertexts in the remaining ballots are homomorphically combined,[21] the homomorphic combination is decrypted to reveal the election outcome, and correctness of decryption is proved in zero-knowledge. Finally, the election outcome and proof of correct decryption are output.

The original Helios scheme [AMPQ09] is vulnerable to attacks against ballot secrecy [CS13, CS11, SC11]. The current version of Helios is intended to mitigate against these attacks.[22] In particular, it incorporates Smyth's recommendation to reject ballots containing zero-knowledge proofs that have been previously observed [Smy12, §4]. For clarity, we write *Helios 3.1.4* for the current version of Helios. Bernhard [Ber14, §6.11] and Bernhard *et al.* [BCG+15a, §D.3] show that variants of Helios 3.1.4 using the strong Fiat-Shamir heuristic satisfy notions of ballot secrecy.[23] These notions assume ballots are *recorded-as-cast*, i.e., cast ballots are preserved with integrity through the ballot collection process [AN06, §2]. Unfortunately, ballot secrecy is not satisfied without this assumption, because Helios 3.1.4 uses malleable ballots (as do the variants studied by Bernhard [Ber14] and Bernhard *et al.* [BCG+15a]), which are incompatible with ballot secrecy (§5).

Theorem 9. *Helios 3.1.4 does not satisfy ballot secrecy.*

Proof sketch. Suppose an adversary calls the left-right oracle to derive a ballot, exploits malleability to derive a related ballot, and outputs a bulletin board containing the related ballot.[24] The board is balanced, because it does not

---

[19] `https://www.iacr.org/elections/`, accessed 1 Sep 2015.

[20] `https://princeton.heliosvoting.org/`, accessed 1 Sep 2015.

[21] The homomorphic combination of ciphertexts is straightforward for two-candidate elections [CF85, BY86, SK94, Ben96, HS00], since choices (e.g., "yes" or "no") can be encoded as 1 or 0. Multi-candidate elections are also possible [BY86, Hir10, DJN10].

[22] `https://github.com/benadida/helios-server/releases/tag/v3.1.4`, accessed 19 Aug 2015.

[23] Bernhard, Pereira & Warinschi [BPW12a] consider whether Helios satisfies a notion of ballot secrecy in two candidate elections.

[24] The recorded-as-cast assumption is violated because the ballot output by the left-right oracle does not appear on the bulletin board.

contain the ballot output by the left-right oracle. And the election outcome will allow the adversary to win the game. □

We omit a formal proof of Theorem 9.

The proof sketch of Theorem 9 does not immediately give way to a real-world attack against Helios. Nevertheless, we can derive an attack (as the following example demonstrates) by extrapolating from the proof sketch and Cortier & Smyth's *permutation attack*, which asserts: given a ballot b for vote v, we can exploit malleability to derive a ballot b' for vote v' [CS13, §3.2.2]. Suppose Alice, Bob and Charlie are voters, and Mallory is an adversary that wants to convince herself that Alice did not vote for a candidate v. Further suppose Alice casts a ballot b_1 for vote v_1, Bob casts a ballot b_2, and Charlie casts a ballot b_3. Moreover, suppose that either Bob or Charlie voted for v. (Thus, we exclude election outcomes without any votes for candidate v, which would permit Mallory to trivially convince herself that Alice did not vote for candidate v.) Let us assume that votes for v' are not expected. Mallory proceeds as follows: she intercepts ballot b_1, exploits malleability to derive a ballot b such that $v = v_1$ implies b is a vote for v', and casts ballot b. It follows that the tallier will compute the election outcome from bulletin board $\{b, b_2, b_3\}$. If the outcome does not contain any votes for v', then Mallory is convinced that Alice did not vote for v. This attack also works against the variants of Helios 3.1.4 studied by Bernhard and Bernhard *et al.*, however, neither Bernhard [Ber14, §6.11] nor Bernhard *et al.* [BCG+15a, §D.3] were able to detect the attack,[25] because interception is not possible when ballots are recorded-as-cast.[26]

Recommendation: adopt non-malleable ballots. We have seen that non-malleable ballots are necessary for ballot secrecy (§5), hence, future Helios releases should adopt non-malleable ballots. The specification for the next Helios release [Adi14] makes some progress in this direction. Moreover, a liberal interpretation of that specification by Smyth, Frink & Clarkson [SFC15] leads to a variant of Helios, named *Helios 4.0*, which defines non-malleable ballots [SHM15]. Proving whether Helios 4.0 satisfies ballot secrecy is a direction for future work. And a successful proof would provide strong motivation for future Helios releases being based upon Helios 4.0.

7 Related work

Discussion of ballot secrecy originates from Chaum [Cha81] and the earliest cryptographic definitions of ballot secrecy are due to Benaloh *et al.* [BY86, BT94, Ben96].[27] More recently, Bernhard *et al.* propose a series of ballot secrecy

[25] Nor could the attack have been detected by Bernhard, Pereira & Warinschi [BPW12a].

[26] This observation suggests that recorded-as-cast is unsatisfiable: an adversary that can intercept ballots can always prevent the collection of ballots. Nevertheless, the definition of recorded-as-cast is informal, thus ambiguity should be expected and some interpretation of the definition should be satisfiable.

[27] Bernhard *et al.* [BCG+15b, BCG+15a] survey ballot secrecy definitions.

definitions: they consider election schemes without tallying proofs [BCP+11, BPW12b] and, subsequently, schemes with tallying proofs [BPW12a, SB13, SB14, BCG+15b]. The definition of ballot secrecy by Bernhard, Pereira & Warinschi computes tallying proofs using algorithm Tally or a simulator [BPW12a], but the resulting definition is too weak [BCG+15b, §3.4] and some strengthening is required [BCG+15b, §4]. By comparison, the definition by Smyth & Bernhard computes tallying proofs using only algorithm Tally [SB13], but the resulting definition is too strong [BCG+15b, §3.5] and a weakening is required [SB14]. The relative merits of ballot secrecy definitions due to Smyth & Bernhard [SB14, Definition 5] and Bernhard *et al.* [BCG+15b, Definition 7] are unknown, in particular, it is unknown whether one definition is stronger than the other.

Discussion of ballot independence originates from Gennaro [Gen95]. And the apparent relationship between ballot secrecy and ballot independence has been considered. Benaloh [Ben96, §2.9] 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. And, more generally, Sako & Kilian [SK95, §2.4], Michels & Horster [MH96, §3] and Cortier & Smyth [CS13, CS11] discuss how non-malleable ballots can be exploited to compromise ballot secrecy. The first cryptographic definition of ballot independence seems to be due to Smyth & Bernhard [SB13, SB14]. Moreover, Smyth & Bernhard formally prove relations between their definitions of ballot secrecy and ballot independence.

All of the ballot secrecy definitions by Bernhard *et al.* [BCP+11, BPW12b, BPW12a, SB13, SB14, BCG+15b] and the ballot independence definition by Smyth & Bernhard [SB13, SB14] focus on detecting attacks by adversaries that control some voters. Attacks by adversaries that control the bulletin board or communication channel are not detected, i.e., the bulletin board is implicitly assumed to operate in accordance with the election scheme's rules and the communication channel is implicitly assumed to be secure. This introduces a trust assumption. Under this assumption, Smyth & Bernhard prove that non-malleable ballots are not necessary for ballot secrecy [SB13, §4.3], and Bernhard [Ber14] and Bernhard *et al.* [BCG+15a, BCG+15b] prove that variants of Helios 3.1.4 satisfy notions of ballot secrecy, despite the use of malleable ballots. By comparison, we prove that non-malleable ballots are necessary for ballot secrecy, without this trust assumption. Hence, the aforementioned variants of Helios 3.1.4 do not satisfy our definition of ballot secrecy. Thus, our definition of ballot secrecy improves upon definitions due to Bernhard *et al.* by detecting more attacks.

Some of the ideas presented in this paper previously appeared in a preliminary unpublished draft by Smyth [Smy14] and, a similarly unpublished, extended version of that draft by Bernhard & Smyth [BS15]. In particular, the limitations of ballot secrecy definitions by Bernhard *et al.* were identified by Smyth [Smy14]. And Definition 2 is based upon the definition of ballot secrecy proposed by Smyth [Smy14, Definition 3]. The main distinction between Definition 2 and the definition by Smyth is syntax: this paper adopts syntax for election schemes from Smyth, Frink & Clarkson [SFC15], whereas, Smyth adopts syntax by Smyth & Bernhard [SB14, SB13]. The change in syntax is motivated by the superiority of syntax by Smyth, Frink & Clarkson. Unfortunately, the change has a drawback: we cannot immediately prove that the definition of ballot secrecy proposed in this paper is strictly stronger than the definition proposed by Smyth & Bernhard [SB14, SB13]. By comparison, the unpublished drafts contain such proofs. Nevertheless, the advantages of the syntax change outweigh the disadvantages.

8 Conclusion

This work was initiated by a desire to eliminate the trust assumptions placed upon the bulletin board and the communication channel in definitions of ballot secrecy by Bernhard *et al.* and the definition of ballot independence by Smyth & Bernhard. This necessitated the introduction of new security definitions.

The definition of ballot secrecy was largely constructed from intuition, with inspiration from indistinguishability games for asymmetric encryption and existing definitions of ballot secrecy. Moreover, the definition was guided by the desire to strengthen existing definitions of ballot secrecy.

The definition of ballot independence was inspired by the realisation that independence essentially states that ballots are non-malleable. This enabled definitions of ballot independence to be constructed as straightforward adaptations of non-malleability and indistinguishability definitions for asymmetric encryption; the former adaptation being a more natural formulation of ballot independence and the latter being simpler.

Relationships between security definitions aid our understanding and offer insights that facilitate the construction of secure election schemes. This prompted the study of relations between ballot secrecy and ballot independence, resulting in a proof that non-malleable ballots are necessary for ballot secrecy. Moreover, a separation result demonstrates that ballot secrecy is strictly stronger than ballot independence.

In light of the revelation that non-malleable ballots are necessary for ballot secrecy and in the knowledge that Helios ballots are malleable, it was discovered that Helios does not satisfy ballot secrecy. Although the proof sketch of this result did not immediately uncover a real-world attack against Helios, an extrapolation from that proof sketch revealed an attack that allows an adversary to determine if a voter did not vote for the adversary's chosen candidate. This naturally led to a consideration of whether definitions of ballot secrecy by Bernhard *et al.* could have detected this attack and the conclusion that they could not, because the attack requires the adversary to control the bulletin board or communication channel, which is prohibited by their definitions.

Acknowledgements. Some of the prose (in particular, the two opening paragraphs of Section 3) were prepared in collaboration with David Bernhard and I am very grateful for David's contribution. I am also grateful to David for extensive discussions that helped improve this paper and, more generally, my knowledge of cryptography. In addition, I am grateful to Elizabeth Quaglia for her valuable feedback that also helped improve this paper. This work was performed in part at INRIA, with support from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC project *CRYSP* (259639).

A Asymmetric encryption

Definition 6 (Asymmetric encryption scheme [KL07]). *An* asymmetric encryption scheme *is a tuple of probabilistic polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, such that:*

- **Gen***, denoted* $(pk, sk) \leftarrow \mathsf{Gen}(\kappa)$*, inputs a security parameter* κ *and outputs a key pair* (pk, sk)*.*
- **Enc***, denoted* $c \leftarrow \mathsf{Enc}(pk, m)$*, inputs a public key* pk *and message* 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 *from the scheme's message space, we have* $\Pr[(pk, sk) \leftarrow \mathsf{Gen}(\kappa); c \leftarrow \mathsf{Enc}(pk, m) : \mathsf{Dec}(sk, c) = m] > 1 - \mathsf{negl}(n)$*. A scheme has* perfect correctness *if the probability is* 1*.*

Definition 7 ($\mathsf{IND\text{-}PA0}$ [BS99]). *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.*

$\mathsf{IND\text{-}PA0}(\Pi, \mathcal{A}, \kappa) =$
$(pk, sk) \leftarrow \mathsf{Gen}(\kappa);$
$(m_0, m_1) \leftarrow \mathcal{A}(pk);$
$\beta \leftarrow_R \{0, 1\};$
$c \leftarrow \mathsf{Enc}(pk, m_\beta);$
$\mathbf{c} \leftarrow \mathcal{A}(c);$
$\mathbf{m} \leftarrow (\mathsf{Dec}(sk, \mathbf{c}[1]), \ldots, \mathsf{Dec}(sk, \mathbf{c}[|\mathbf{c}|]);$
$g \leftarrow \mathcal{A}(\mathbf{m});$
return $g = \beta \wedge \bigwedge_{1 \leq i \leq |\mathbf{c}|} c \neq \mathbf{c}[i];$

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

B Proofs

B.1 Proof of Proposition 3

We present a construction (Definition 8) for encryption schemes (Lemma 10) which are clearly not secure (Lemma 11). Nevertheless, the construction produces encryption schemes that are sufficient for ballot secrecy (Lemma 12). The proof of Proposition 3 follows from Lemmata 10–12.

Definition 8. *Given an asymmetric encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *and a constant symbol* ω*, let* $\mathsf{Leak}(\Pi, \omega) = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}')$*, such that* $\mathsf{Dec}'(sk, c)$ *proceeds as follows: if* $c = \omega$*, then output* sk*, otherwise, compute* $m \leftarrow \mathsf{Dec}(sk, c)$ *and output* m*.*

Lemma 10. *Given an asymmetric encryption scheme* Π *and a constant symbol* ω*, such that* Π*'s ciphertext space does not contain* ω*, we have* $\mathsf{Leak}(\Pi, \omega)$ *is an asymmetric encryption scheme.*

Proof sketch. The proof follows immediately from correctness of the underlying encryption scheme, because constant symbol ω does not appear in the scheme's ciphertext space. $\square$

Lemma 11. *Given an asymmetric encryption scheme* Π *and a constant symbol* ω*, such that* Π*'s ciphertext space does not contain* ω *and* Π*'s message space is larger than one for some security parameter, we have* $\mathsf{Leak}(\Pi, \omega)$ *does not satisfy* $\mathsf{IND\text{-}PA0}$*.*

Proof sketch. The proof is trivial: an adversary can output two distinct messages and a vector containing constant symbol ω during the first two adversary calls, learn the private key from the parallel decryption, and use the key to recover the plaintext from the challenge ciphertext, which allows the adversary to win the game. $\square$

Lemma 12. *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an asymmetric encryption scheme and* ω *be a constant symbol. Suppose* Π*'s ciphertext space does not contain* ω *and* Π*'s message space is smaller than the private key. Further suppose* $\mathsf{Enc2Vote}(\Pi)$ *satisfies* $\mathsf{Ballot\text{-}Secrecy}$*. We have* $\mathsf{Enc2Vote}(\mathsf{Leak}(\Pi, \omega))$ *satisfies* $\mathsf{Ballot\text{-}Secrecy}$*.*

Proof. Let $\mathsf{Enc2Vote}(\Pi) = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{Tally})$ and let $\mathsf{Enc2Vote}(\mathsf{Leak}(\Pi, \omega)) = (\overline{\mathsf{Setup}}, \overline{\mathsf{Vote}}, \overline{\mathsf{Tally}})$. By definition of $\mathsf{Enc2Vote}$ and Leak, we have $\mathsf{Setup} = \overline{\mathsf{Setup}}$ and $\mathsf{Vote} = \overline{\mathsf{Vote}}$. Suppose $\mathfrak{m}$ is Π's message space. By definition of Leak, we have $\mathfrak{m}$ is $\mathsf{Leak}(\Pi, \omega)$'s message space too. Moreover, since $|\mathfrak{m}|$ is smaller than the private key, we have for all security parameters κ, bulletin boards $\mathfrak{bb}$, and number of candidates nc, that $nc \leq |\mathfrak{m}|$ implies

$$\begin{aligned}\Pr[(pk, sk) \leftarrow \mathsf{Gen}(\kappa); (\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, nc, \mathfrak{bb}, \kappa);\\ (\overline{\mathfrak{v}}, \overline{pf}) \leftarrow \overline{\mathsf{Tally}}(sk, nc, \mathfrak{bb}, \kappa) : \mathfrak{v} = \overline{\mathfrak{v}} \wedge pf = \overline{pf}] = 1,\end{aligned}$$

because $\mathsf{Enc2Vote}$ ensures that $\overline{\mathfrak{v}}$ is not influenced by decrypting ω (witness that decrypting ω outputs sk such that $sk > |\mathfrak{m}| \geq nc$) and pf is a constant symbol. It follows for all adversaries $\mathcal{A}$ and security parameters κ that games $\mathsf{Ballot\text{-}Secrecy}(\mathsf{Enc2Vote}(\Pi), \mathcal{A}, \kappa)$ and $\mathsf{Ballot\text{-}Secrecy}(\mathsf{Enc2Vote}(\mathsf{Leak}(\Pi, \omega)), \mathcal{A}, \kappa)$ are equivalent, hence, we have $\mathsf{Succ}(\mathsf{Ballot\text{-}Secrecy}(\mathsf{Enc2Vote}(\Pi), \mathcal{A}, \kappa)) = \mathsf{Succ}(\mathsf{Ballot\text{-}Secrecy}(\mathsf{Enc2Vote}(\mathsf{Leak}(\Pi, \omega)), \mathcal{A}, \kappa))$. Moreover, since $\mathsf{Enc2Vote}(\Pi)$ satisfies $\mathsf{Ballot\text{-}Secrecy}$, it follows that $\mathsf{Enc2Vote}(\mathsf{Leak}(\Pi, \omega))$ satisfies $\mathsf{Ballot\text{-}Secrecy}$ too. □

Proof of Proposition 3. Let Π be an asymmetric encryption scheme and ω be a constant symbol. Suppose Π's ciphertext space does not contain ω. Further suppose Π's message space is larger than one for some security parameter, but smaller than the private key. We have $\mathsf{Enc2Vote}(\mathsf{Leak}(\Pi, \omega))$ is an asymmetric encryption scheme (Lemma 10) such that $\mathsf{Enc2Vote}(\mathsf{Leak}(\Pi, \omega))$ satisfies $\mathsf{Ballot\text{-}Secrecy}$ (Lemma 12), but $\mathsf{Leak}(\Pi, \omega)$ does not satisfy $\mathsf{IND\text{-}PA0}$ (Lemma 11), concluding our proof. □

B.2 Proof of Theorem 4

For the *if* implication, suppose Γ does not satisfy $\mathsf{CNM\text{-}CVA}$, hence, there exists a probabilistic polynomial-time adversary $\mathcal{A}$, such that for all negligible functions negl, there exists a security parameter κ, and $\mathsf{Succ}(\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa)) > \mathsf{negl}(\kappa)$. We construct an adversary $\mathcal{B}$ against game $\mathsf{IND\text{-}CVA}$ from adversary $\mathcal{A}$.

- $\mathcal{B}(pk, \kappa)$ computes $(V, nc) \leftarrow \mathcal{A}(pk, \kappa); v, v' \leftarrow_R V$ and outputs (v, v', nc).
- $\mathcal{B}(b)$ computes $(R, \mathfrak{bb}) \leftarrow \mathcal{A}(b)$ and outputs $\mathfrak{bb}$.
- $\mathcal{B}(\mathfrak{v})$ outputs 0 if $R(v, \mathfrak{v})$ holds and 1 otherwise.

If the challenger selects $\beta = 0$ in $\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{B}, \kappa)$, then adversary $\mathcal{B}$ simulates $\mathcal{A}$'s challenger to $\mathcal{A}$ in $\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa)$ and $\mathcal{B}$'s success (which requires $R(v, \mathfrak{v})$ to hold) is $\mathsf{Succ}(\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa))$. Otherwise ($\beta = 1$), adversary $\mathcal{B}$ simulates $\mathcal{A}$'s challenger to $\mathcal{A}$ in $\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa)$ and, since $\mathcal{B}$ will evaluate $R(v, \mathfrak{v})$, $\mathcal{B}$'s success (which requires $R(v, \mathfrak{v})$ not to hold) is $1 - \mathsf{Succ}(\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa))$. It follows that $\mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{A}, \kappa)) = 1/2 \cdot (\mathsf{Succ}(\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa)) + 1 - \mathsf{Succ}(\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa)))$ and, therefore, $2 \cdot \mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{A}, \kappa)) - 1 = \mathsf{Succ}(\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa))$. Since Γ does not satisfy $\mathsf{CNM\text{-}CVA}$ and a function that doubles the output of a negligible function is a negligible function, we have $\mathsf{Succ}(\mathsf{cnm\text{-}cva}(\Gamma, \mathcal{A}, \kappa)) - \mathsf{Succ}(\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma, \mathcal{A}, \kappa)) > 2 \cdot \mathsf{negl}(\kappa)$. It follows that $2 \cdot \mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{A}, \kappa)) - 1 > 2 \cdot \mathsf{negl}(\kappa)$, hence, $\mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{A}, \kappa)) > 1/2 + \mathsf{negl}(\kappa)$, concluding our proof.

For the *only if* implication, suppose Γ does not satisfy $\mathsf{IND\text{-}CVA}$, hence, there exists a probabilistic polynomial-time adversary $\mathcal{A}$, such that for all negligible functions negl, there exists a security parameter κ, and $\mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma, \mathcal{A}, \kappa)) > 1/2 + \mathsf{negl}(\kappa)$. We construct an adversary $\mathcal{B}$ against $\mathsf{CNM\text{-}CVA}$ from adversary $\mathcal{A}$.

- $\mathcal{B}(pk,\kappa)$ computes $(v_0, v_1, nc) \leftarrow \mathcal{A}(pk,\kappa)$ and outputs $(\{v_0, v_1\}, nc)$.
- $\mathcal{B}(b)$ computes $\mathfrak{bb} \leftarrow \mathcal{A}(b)$, picks coins r uniformly at random, derives a relation R such that $R(v, \mathfrak{v})$ holds if there exists a bit g such that $v = v_g \wedge g = \mathcal{A}(\mathfrak{v}; r)$ and fails otherwise, and outputs $(R, \mathfrak{bb})$.

Adversary $\mathcal{B}$ simulates $\mathcal{A}$'s challenger to $\mathcal{A}$ in game $\mathsf{IND\text{-}CVA}(\Gamma,\mathcal{A},\kappa)$. Indeed, the challenge ballot is equivalently computed. As is the election outcome. The computation $\mathcal{A}(\mathfrak{v}; r)$ is not black-box, but this does not matter: it is still invoked exactly one time in the game. Let use consider adversary $\mathcal{B}$'s success in $\mathsf{cnm\text{-}cva}(\Gamma,\mathcal{B},\kappa)$ and $\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma,\mathcal{B},\kappa)$.

- Game $\mathsf{cnm\text{-}cva}(\Gamma,\mathcal{B},\kappa)$ samples a single vote v from V. By inspection of $\mathsf{cnm\text{-}cva}(\Gamma,\mathcal{B},\kappa)$ and $\mathsf{IND\text{-}CVA}(\Gamma,\mathcal{A},\kappa)$, we have $\mathsf{Succ}(\mathsf{cnm\text{-}cva}(\Gamma,\mathcal{B},\kappa)) = \mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma,\mathcal{A},\kappa))$, hence, $\mathsf{Succ}(\mathsf{cnm\text{-}cva}(\Gamma,\mathcal{B},\kappa)) - 1/2 > \mathsf{negl}(\kappa)$.
- Game $\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma,\mathcal{B},\kappa)$ samples votes v and v' from V. Vote v is independent of $\mathcal{A}$'s perspective, indeed, an equivalent formulation of $\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma,\mathcal{B},\kappa)$ could sample v after $\mathcal{A}$ has terminated and immediately before evaluating the adversary's relation. By inspection of $\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma,\mathcal{B},\kappa)$ and $\mathsf{IND\text{-}CVA}(\Gamma,\mathcal{A},\kappa)$, we have $\mathsf{Succ}(\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma,\mathcal{B},\kappa)) = 1/2 \cdot \mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma,\mathcal{A},\kappa)) + 1/2 \cdot (1 - \mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma,\mathcal{A},\kappa))) = 1/2$.

It follows that $\mathsf{Succ}(\mathsf{cnm\text{-}cva}(\Gamma,\mathcal{B},\kappa)) - \mathsf{Succ}(\mathsf{cnm\text{-}cva\text{-}\$}(\Gamma,\mathcal{B},\kappa)) > \mathsf{negl}(\kappa)$. $\square$

B.3 Proof of Theorem 7

Suppose Γ does not satisfy ballot independence, hence, there exists a probabilistic polynomial-time adversary $\mathcal{A}$, such that for all negligible functions negl, there exists a security parameter κ, and $\mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma,\mathcal{A},\kappa)) > 1/2 + \mathsf{negl}(\kappa)$. We construct a ballot secrecy adversary $\mathcal{B}$ from the ballot independence adversary $\mathcal{A}$.

- $\mathcal{B}(pk,\kappa)$ computes $(v_0, v_1, nc) \leftarrow \mathcal{A}(pk,\kappa)$ and outputs nc.
- $\mathcal{B}()$ computes $b \leftarrow \mathcal{O}(v_0, v_1); \mathfrak{bb} \leftarrow \mathcal{A}(b)$ and outputs $\mathfrak{bb}$.
- $\mathcal{B}(\mathfrak{v}, pf)$ computes $g \leftarrow \mathcal{A}(\mathfrak{v})$ and outputs g.

Adversary $\mathcal{B}$ simulates $\mathcal{A}$'s challenger to $\mathcal{A}$. Indeed, the challenge ballot and election outcome are equivalently computed. Moreover, the challenge ballot does not appear on the bulletin board, hence, the bulletin board is balanced. It follows that $\mathsf{Succ}(\mathsf{IND\text{-}CVA}(\Gamma,\mathcal{A},\kappa)) = \mathsf{Succ}(\mathsf{Ballot\text{-}Secrecy}(\Gamma,\mathcal{B},\kappa))$, hence, $\mathsf{Succ}(\mathsf{Ballot\text{-}Secrecy}(\Gamma,\mathcal{B},\kappa)) > 1/2 + \mathsf{negl}(\kappa)$, concluding our proof. $\square$

References

[Adi09] Ben Adida. Helios deployed at Princeton. http://heliosvoting.wordpress.com/2009/10/13/helios-deployed-at-princeton/ (accessed 7 May 2014), 2009.

[Adi14] Ben Adida. Helios v4 Verification Specs. Helios documentation, http://documentation.heliosvoting.org/verification-specs/helios-v4 (accessed 2 May 2014), 2014. A snapshot of the specification on 18 Oct 2013 is available from https://web.archive.org/web/20131018033747/http://documentation.heliosvoting.org/verification-specs/helios-v4.

[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.

[AN06] Ben Adida and C. Andrew Neff. Ballot casting assurance. In *EVT'06: Electronic Voting Technology Workshop*, Berkeley, CA, USA, 2006. USENIX Association.

[BCG+15a] David Bernhard, Véronique Cortier, David Galindo, Olivier Pereira, and Bogdan Warinschi. A comprehensive analysis of game-based ballot privacy definitions. Cryptology ePrint Archive, Report 2015/255 (version 20150319:100626), 2015.

[BCG+15b] David Bernhard, Véronique Cortier, David Galindo, Olivier Pereira, and Bogdan Warinschi. SoK: A comprehensive analysis of game-based ballot privacy definitions. In *S&P'15: 36th Security and Privacy Symposium*. IEEE Computer Society, 2015.

[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.

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A Concrete Security Treatment of Symmetric Encryption. In *FOCS'97: 38th Annual Symposium on Foundations of Computer Science*, pages 394–403. IEEE Computer Society, 1997.

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

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

[Ber14] David Bernhard. *Zero-Knowledge Proofs in Theory and Practice*. PhD thesis, Department of Computer Science, University of Bristol, 2014.

[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.

[Bow07] Debra 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://admin.cdn.sos.ca.gov/press-releases/prior/2007/DB07_111.pdf` (accessed 1 September 2015), August 2007. A snapshot of the press release on 6 February 2008 is available from `https://web.archive.org/web/20080206210142/http://www.sos.ca.gov/elections/voting_systems/ttbr/db07_042_ttbr_system_decisions_release.pdf`.

[BPW12a] David Bernhard, Olivier Pereira, and Bogdan 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.

[BPW12b] David Bernhard, Olivier Pereira, and Bogdan Warinschi. On Necessary and Sufficient Conditions for Private Ballot Submission. Cryptology ePrint Archive, Report 2012/236 (version 20120430:154117b), 2012.

[BR05] Mihir Bellare and Phillip Rogaway. Symmetric Encryption. In *Introduction to Modern Cryptography*, chapter 4. 2005. `http://cseweb.ucsd.edu/~mihir/cse207/w-se.pdf`. A snapshot of the chapter on 21 Mar 2015 is available from `https://web.archive.org/web/20150321170845/http://cseweb.ucsd.edu/~mihir/cse207/w-se.pdf`.

[BS99] Mihir Bellare and Amit 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.

[BS15] David Bernhard and Ben Smyth. Ballot secrecy with malicious bulletin boards. Cryptology ePrint Archive, Report 2014/822 (version 20150413:170300), 2015.

[BT94] Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In *STOC'94: 26th Theory of computing Symposium*, pages 544–553. ACM Press, 1994.

[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.

[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.

[Cha81] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84–90, 1981.

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

[CS13] Véronique Cortier and Ben Smyth. Attacking and fixing Helios: An analysis of ballot secrecy. *Journal of Computer Security*, 21(1):89–148, 2013.

[DC12] Yvo Desmedt and Pyrros Chaidos. Applying Divertibility to Blind Ballot Copying in the Helios Internet Voting System. In *ESORICS'12: 17th European Symposium on Research in Computer Security*, volume 7459 of *LNCS*, pages 433–450. Springer, 2012.

[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.

[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.

[DK05] Yvo Desmedt and Kaoru Kurosawa. Electronic Voting: Starting Over? In *ISC'05: International Conference on Information Security*, volume 3650 of *LNCS*, pages 329–343. Springer, 2005.

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

[GH07] Rop Gonggrijp and Willem-Jan Hengeveld. Studying the Nedap/Groenendaal ES3B Voting Computer: A Computer Security Perspective. In *EVT'07: Electronic Voting Technology Workshop*, 2007.

[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.

[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.

[HK02] Alejandro Hevia and Marcos A. Kiwi. Electronic Jury Voting Protocols. In *LATIN'02: Theoretical Informatics*, volume 2286 of *LNCS*, pages 415–429. Springer, 2002.

[HK04] Alejandro Hevia and Marcos A. Kiwi. Electronic jury voting protocols. *Theoretical Computer Science*, 321(1):73–94, 2004.

[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.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography.* Chapman & Hall/CRC, 2007.

[MH96] Markus Michels and Patrick 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.

[OAS69] American Convention on Human Rights, "Pact of San Jose, Costa Rica", 1969.

[OSC90] Document of the Copenhagen Meeting of the Conference on the Human Dimension of the CSCE, 1990.

[QS15] Elizabeth A. Quaglia and Ben Smyth. Constructing secret, verifiable auction schemes from election schemes. Unpublished draft, 2015.

[SB13] Ben Smyth and David 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.

[SB14] Ben Smyth and David Bernhard. Ballot secrecy and ballot independence: definitions and relations. Cryptology ePrint Archive, Report 2013/235 (version 20141010:082554), 2014.

[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/`.

[SFC15] Ben Smyth, Steven Frink, and Michael R. Clarkson. Election Verifiability: Cryptographic Definitions and an Analysis of Helios and JCJ. Cryptology ePrint Archive, Report 2015/233 (version 20150807:085847), 2015.

[SFD+14] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Alex 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.

[SHM15] Ben Smyth, Yoshikazu Hanatani, and Hirofumi Muratani. NM-CPA secure encryption with proofs of plaintext knowledge. In *IWSEC'15: 10th International Workshop on Security*, volume 9241 of *LNCS*. Springer, 2015.

[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.

[SK95] Kazue Sako and Joe 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.

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

[Smy14] Ben Smyth. Ballot secrecy with malicious bulletin boards. Cryptology ePrint Archive, Report 2014/822 (version 20141012:004943), 2014.

[UN48] Universal Declaration of Human Rights, 1948.

[WWH+10] Scott Wolchok, Eric Wustrow, J. Alex Halderman, Hari K. Prasad, Arun Kankipati, Sai Krishna Sakhamuri, Vasavya Yagati, and Rop 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.

[WWIH12] Scott Wolchok, Eric Wustrow, Dawn Isabel, and J. Alex 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.