Ballot Secrecy and Ballot Independence Coincide*

Ben Smyth[1] and David Bernhard[2]

[1] INRIA Paris-Rocquencourt, France
[2] University of Bristol, England

Abstract. We study ballot independence for election schemes:

- We formally define ballot independence as a cryptographic game and prove that ballot secrecy implies ballot independence.
- We introduce a notion of controlled malleability and show that it is sufficient for ballot independence. We also show that non-malleable ballots are sufficient, but not necessary, for ballot independence.
- We prove that ballot independence is sufficient for ballot secrecy under practical assumptions.

Our results show that ballot independence is necessary in election schemes satisfying ballot secrecy. Furthermore, our sufficient conditions enable simpler proofs of ballot secrecy.

1 Introduction

Voters should be able to express their free will in elections without fear of retribution; this property is known as privacy. Cryptographic formulations of privacy depend on the specific setting and *ballot secrecy*[1] [2–4] has emerged as a *de facto* standard privacy requirement of election schemes.

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

Ballot secrecy provides privacy in an intimidation-free environment and stronger properties such as *receipt-freeness* and *coercion resistance* [5] provide privacy in environments where intimidation may occur. Bernhard *et al.* [6–8] propose a cryptographic formalisation of ballot secrecy. However, we show that their definition allows election schemes that reveal voters' votes to be proven secure and we strengthen the definition to prevent this issue.

Ballot independence [4, 9] is seemingly related to ballot secrecy.

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

[*] The full version of this paper is available as an IACR Cryptology ePrint [1].

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

J. Crampton, S. Jajodia, and K. Mayes (Eds.): ESORICS 2013, LNCS 8134, pp. 463–480, 2013.

Indeed, Cortier and Smyth [4, 10, 11] attribute a class of ballot secrecy attacks to the absence of ballot independence. However, ballot independence has not been formally defined and its relationship with ballot secrecy is unknown. We provide a definition of ballot independence and show that ballot secrecy and ballot independence coincide in practical settings.

In traditional paper-based elections, physical mechanisms can be used to achieve privacy, for instance, ballots are completed in isolation inside polling booths, placed into locked ballot boxes, and mixed with other ballots before tallying. (See Schneier [12] for a detailed, informal security analysis of Papal elections.) By comparison, the provision of ballot secrecy is more difficult in end-to-end verifiable election schemes, since ballots are posted on publicly readable bulletin boards. Nonetheless, ballot secrecy is a *de facto* standard property of election schemes and, hence, must be satisfied. The aforementioned physical mechanisms also provide an assurance of ballot independence in paper-based elections, however, the motivation for election schemes satisfying ballot independence is unclear, indeed, Bulens, Giry & Pereira [13, §3.2] question whether ballot independence is a desirable property of election schemes and highlight the investigation of voting schemes which allow the submission of related votes whilst preserving ballot secrecy as an interesting research direction. Moreover, in the context of the Helios [14, 15] election scheme, Desmedt & Chaidos [16] present a protocol which allows Bob to cast the same vote as Alice, with Alice's cooperation, and claim that Bob cannot learn Alice's vote. In this paper, we study the relationship between ballot secrecy and ballot independence and show that the two properties coincide in practical settings.

Contribution and Outline. In Section 3 we show that the definition of ballot secrecy by Bernhard *et al.* allows election schemes that reveal voters' votes to be proven secure and we present a stronger definition of ballot secrecy to prevent this issue. In Section 4 we propose a definition of ballot independence and give sufficient conditions to achieve this notion, including a definition of controlled-malleable encryption. In Section 5 we prove that ballot secrecy implies ballot independence, thereby providing an argument to end the ballot independence debate: ballot independence is a necessary property of election schemes (assuming ballot secrecy is required). In addition, we critique (Section 5.1) the results by Desmedt & Chaidos and argue that their security results do not support their claims. In Section 6 we present a practical class of election schemes (which includes Helios) for which ballot secrecy and ballot independence coincide.

Related work. The concept of independence was introduced by Chor *et al.* [17] and studied in the context of election schemes by Gennaro [9]. Cortier and Smyth [4, 10, 11] have discovered attacks on ballot secrecy in several election schemes and considered the relationship to independence [4, Section 7]; their evidence suggests ballot secrecy implies ballot independence in homomorphic voting systems such as Helios. However, Cortier & Smyth did not make any formal claims, because ballot independence had not been formally defined. By comparison, in this paper, we present a formal definition of ballot independence and prove that ballot secrecy implies ballot independence. Bernhard, Pereira & Warinschi [7] show that a non-malleable encryption scheme is sufficient to build an election scheme satisfying ballot secrecy and our work generalises their result.

2 Preliminaries

We adopt standard notation for the application of probabilistic algorithms: if A is a probabilistic algorithm, then $A(x_1, \ldots, x_n; r)$ is the result of running A on input $x_1, \ldots, x_n$ and coins r. We let $y \leftarrow A(x_1, \ldots, x_n)$ denote picking r at random and assigning the output of $A(x_1, \ldots, x_n; r)$ to the variable y. If S is a finite set, then $x \leftarrow S$ assigns a uniformly chosen element of S to x. If α is neither a probabilistic algorithm nor a set, then $x \leftarrow \alpha$ assigns α to x. Vectors are denoted using boldface, for example, $\mathbf{x}$. We extend set membership notation to vectors: we write $x \in \mathbf{x}$ (respectively, $x \notin \mathbf{x}$) if x is an element (respectively, x is not an element) of the vector $\mathbf{x}$.

2.1 Non-malleable Encryption

Let us recall the standard syntax for *asymmetric encryption schemes.*

Definition 1 (Asymmetric encryption scheme). *An* asymmetric encryption scheme *is a triple of efficient algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

- *The* key generation algorithm Gen *takes a security parameter* 1^n *as input and outputs a key pair* (pk, sk)*, where* pk *is a public key and* sk *is a private key.*
- *The* encryption algorithm Enc *takes a public key* pk *and message* m *as input, and outputs a ciphertext* c*.*
- *The* decryption algorithm Dec *takes a private key* sk *and ciphertext* c *as input, and outputs a message* m *or the special symbol* $\perp$ *denoting failure.*

Moreover, the scheme must be correct: for all $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$*, we have for all messages* m *and ciphertexts* $c \leftarrow \mathsf{Enc}_{pk}(m)$*, that* $\mathsf{Dec}_{sk}(c) = m$ *with overwhelming probability.*

Non-malleability [18–20] is a standard computational security model used to evaluate the suitability of encryption schemes. Intuitively, if an encryption scheme satisfies non-malleability, then an adversary is unable to construct a ciphertext *"meaningfully related"* to a challenge ciphertext, thereby capturing the idea that ciphertexts are tamper-proof. Formally, Definition 2 recalls the non-malleability game proposed by Bellare *et al.* [19].

Definition 2 (Non-malleable encryption). *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an asymmetric encryption scheme,* $\mathcal{A} = (A_1, A_2)$ *be an adversary, and*

$$\mathsf{NM\text{-}CPA}_{\mathcal{A},\Pi}(n) := |\mathsf{Succ}^{CPA}_{\mathcal{A},\Pi}(n) - \mathsf{Succ}^{CPA}_{\mathcal{A},\Pi,\$}(n)|$$

where $\mathsf{Succ}^{CPA}_{\mathcal{A},\Pi}(n)$ and $\mathsf{Succ}^{CPA}_{\mathcal{A},\Pi,\$}(n)$ are defined below, and n is a security parameter.

$$\begin{aligned}\mathsf{Succ}^{CPA}_{\mathcal{A},\Pi}(n) = Pr[&(pk, sk) \leftarrow \mathsf{Gen}(1^n);\ (M, s) \leftarrow A_1(pk);\\ &x \leftarrow M;\ y \leftarrow \mathsf{Enc}_{pk}(x);\ (R, \mathbf{y}) \leftarrow A_2(M, s, y);\\ &\mathbf{x} \leftarrow \mathsf{Dec}_{sk}(\mathbf{y}) : y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(x, \mathbf{x})]\end{aligned}$$

$$\begin{aligned}\mathsf{Succ}^{CPA}_{\mathcal{A},\Pi,\$}(n) = Pr[&(pk, sk) \leftarrow \mathsf{Gen}(1^n);\ (M, s) \leftarrow A_1(pk);\\ &x, x' \leftarrow M;\ y \leftarrow \mathsf{Enc}_{pk}(x);\ (R, \mathbf{y}) \leftarrow A_2(M, s, y);\\ &\mathbf{x} \leftarrow \mathsf{Dec}_{sk}(\mathbf{y}) : y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(x', \mathbf{x})]\end{aligned}$$

In the above games we insist that the message space is valid (that is, $|x| = |x'|$ for any $x, x' \leftarrow M$ given non-zero probability in the message space) and samplable in polynomial time, and the relation R is computable in polynomial time. We say Π satisfies NM-CPA *if for all probabilistic polynomial-time adversaries $\mathcal{A}$ and security parameters n, there exists a negligible function* negl *such that* $\mathsf{NM\text{-}CPA}_{\mathcal{A},\Pi}(n) \leq \mathsf{negl}(n)$.

3 Election Schemes and Ballot Secrecy

Based upon Bernhard *et al.* [6–8], we define a syntax for *election schemes* as follows.

Definition 3 (Election scheme). *An* election scheme *is a tuple of efficient algorithms* (Setup, Vote, BB, Tally) *such that:*

- *The* setup algorithm Setup *takes a security parameter 1^n as input and outputs a bulletin board $\mathfrak{bb}$, vote space $\mathfrak{m}$, public key pk, and private key sk, where $\mathfrak{bb}$ is a multiset and $\mathfrak{m}$ is a set.*
- *The* vote algorithm Vote *takes a public key pk and vote $v \in \mathfrak{m}$ as input, and outputs a ballot b.*
- *The* bulletin board algorithm BB *takes a bulletin board $\mathfrak{bb}$ and ballot b as input, where $\mathfrak{bb}$ is a multiset. It outputs $\mathfrak{bb} \cup \{b\}$ if successful (i.e., b is added to $\mathfrak{bb}$) or $\mathfrak{bb}$ to denote failure (i.e., b is not added).*
- *The* tally algorithm Tally *takes a private key sk and bulletin board $\mathfrak{bb}$ as input, where $\mathfrak{bb}$ is a multiset. It outputs a multiset $\mathfrak{v}$ representing the election result if successful or the empty set $\emptyset$ to denote failure, and auxiliary data aux.*

Moreover, the scheme must satisfy the following correctness property: for all parameters $(\mathfrak{bb}_0, \mathfrak{m}, pk, sk) \leftarrow \mathsf{Setup}(1^n)$, votes $v \in \mathfrak{m}$, multisets $\mathfrak{bb}$, ballots $b \leftarrow \mathsf{Vote}_{pk}(v)$, bulletin boards $\mathfrak{bb}' \leftarrow \mathsf{BB}(\mathfrak{bb}, b)$ and tallying data $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb})$ and $(\mathfrak{v}', aux') \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}')$, we have with overwhelming probability that $\mathfrak{bb}' = \mathfrak{bb} \cup \{b\}$ and if $\mathfrak{v} \neq \emptyset$, then $\mathfrak{v}' = \mathfrak{v} \cup \{v\}$ and $|\mathfrak{v}| = |\mathfrak{bb}|$, otherwise, $\mathfrak{v}' = \emptyset$.

In comparison with earlier presentations by Bernhard *et al.*, Definition 3 is stricter, since we explicitly define the bulletin board and election result as multisets. Moreover, the correctness condition, asserting that the election result corresponds to the multiset of votes cast, is new. Although the correctness condition restricts the applicability of our definition – for example, we cannot model schemes with weighted votes nor schemes which only reveal the winning candidate (as opposed to the number of votes for each candidate) – we believe it is useful for simplicity. In addition, there are some minor differences in error handling and we merge some functionality into a single function[2].

We demonstrate the applicability of our definition by recalling the construction (Definition 4) for election schemes proposed by Bernhard *et al.* [6,7]. We stress that more sophisticated schemes can also be captured – for example, Bernhard *et al.* [6–8] model Helios – but the following scheme is sufficient for our purposes.

Definition 4 ($\mathsf{Enc2Vote}$). *Given an asymmetric encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, we define the election scheme $\mathsf{Enc2Vote}(\Pi)$ as follows.*

- Setup *takes a security parameter 1^n as input and outputs $(\emptyset, \mathfrak{m}, pk, sk)$, where $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$ and $\mathfrak{m}$ is the encryption scheme's message space.*
- Vote *takes a public key pk and vote $v \in \mathfrak{m}$ as input, and outputs $\mathsf{Enc}_{pk}(v)$.*
- BB *takes a bulletin board $\mathfrak{bb}$ and ballot b as input, where $\mathfrak{bb}$ is a multiset. If $b \in \mathfrak{bb}$, then the algorithm outputs $\mathfrak{bb}$ (denoting failure), otherwise, the algorithm outputs $\mathfrak{bb} \cup \{b\}$.*
- Tally *takes as input a private key sk and a bulletin board $\mathfrak{bb}$, where $\mathfrak{bb}$ is a multiset. It outputs the multiset $\{\mathsf{Dec}_{sk}(b) \mid b \in \mathfrak{bb}\}$ and auxiliary data $\perp$.*

Intuitively, given an asymmetric encryption scheme Π satisfying $\mathsf{NM\text{-}CPA}$, the construction $\mathsf{Enc2Vote}(\Pi)$ derives ballot secrecy from Π until tallying and the Tally algorithm maintains ballot secrecy by returning the number of votes for each candidate as an unordered multiset of votes[3].

Ballot Secrecy. Ballot secrecy is a *de facto* standard property of election schemes and, based upon Bernhard *et al.* [6–8], we formalise a cryptographic game for ballot secrecy (Definition 5). We will describe the differences between

[2] In essence, the tally algorithm defined by Bernhard *et al.* outputs a tally τ and an additional algorithm is used to compute the election result $\mathfrak{v}$ from τ. We combine the functionality of these two algorithms into a single function but distinguish between the result $\mathfrak{v}$ and auxiliary data *aux*, which is typically used to store signatures of knowledge proving that the election result has been correctly computed from the bulletin board.

[3] Definition 4 rectifies a mistake in the presentation by Bernhard, Pereira & Warinschi [7] which outputs a vector of votes (rather than a multiset) ordered by the time at which each vote was cast and therefore does not provide ballot secrecy, since there is a mapping between the order in which votes were cast and the votes. (Bernhard *et al.* [6] avoid this problem in a similar fashion.)

our formalisation and earlier presentations after our definition. Informally, our game proceeds as follows. First, the challenger executes the setup algorithm to construct a bulletin board $\mathfrak{bb}_0$, a vote space $\mathfrak{m}$, a public key pk, and a private key sk; the challenger also initialises a bulletin board $\mathfrak{bb}_1$ as a copy of $\mathfrak{bb}_0$ and selects a random bit β. Secondly, the adversary executes the algorithm A_1. The algorithm A_1 has access to an oracle $\mathcal{O}$ as follows: $\mathcal{O}(v_0, v_1)$ allows the adversary to honestly cast a vote $v_0 \in \mathfrak{m}$ on bulletin board $\mathfrak{bb}_0$ and honestly cast a vote $v_1 \in \mathfrak{m}$ on bulletin board $\mathfrak{bb}_1$, where the votes are cast using ballots constructed by the Vote algorithm; $\mathcal{O}(b)$ allows the adversary to cast a ballot b, where b is constructed by the adversary and might be rejected by the bulletin board; and $\mathcal{O}()$ returns the bulletin board $\mathfrak{bb}_\beta$. Thirdly, the challenger computes the election result $\mathfrak{v}$ as follows: if the honestly cast votes on the bulletin board $\mathfrak{bb}_0$ correspond to the honestly cast votes on the bulletin board $\mathfrak{bb}_1$, then the challenger reveals the election result for $\mathfrak{bb}_\beta$, otherwise, the challenger reveals the election result for $\mathfrak{bb}_0$, thereby preventing the adversary from trivially revealing β when the honestly cast votes differ. (The distinction between $\mathfrak{bb}_0$ and $\mathfrak{bb}_1$ is trivial when the honestly cast votes differ, because the adversary can test for the presence of honestly cast votes in the election result.) Formally, we introduce the multisets L_0 and L_1 to record the honestly cast votes on bulletin boards $\mathfrak{bb}_0$ and $\mathfrak{bb}_1$, and model the correspondence between bulletin boards as an equality test on L_0 and L_1, that is, we compute $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_\alpha)$ such that $\alpha = \beta$, if $L_0 = L_1$, and $\alpha = 0$, otherwise. Finally, the adversary executes the algorithm A_2 on the election result $\mathfrak{v}$ and any state information s provided by A_1. The election scheme satisfies ballot secrecy if the adversary has less than a negligible advantage over guessing the bulletin board she interacted with.

Definition 5 (IND-SEC: Ballot secrecy). *Let $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ be an election scheme, $\mathcal{A} = (A_1, A_2)$ be an adversary, and IND-SEC$_{\mathcal{A},\Gamma}(n)$ be the quantity defined below, where n is the security parameter.*

$$2 \cdot Pr[L_0 \leftarrow \emptyset; L_1 \leftarrow \emptyset; (\mathfrak{bb}_0, \mathfrak{m}, pk, sk) \leftarrow \mathsf{Setup}(1^n);\ \mathfrak{bb}_1 \leftarrow \mathfrak{bb}_0;\ \beta \leftarrow \{0,1\};$$
$$s \leftarrow A_1^{\mathcal{O}}(\mathfrak{m}, pk);\ (\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_\alpha) : A_2(\mathfrak{bb}_\beta, \mathfrak{v}, aux, s) = \beta] - 1$$

In the above game, L_0 and L_1 are multisets, the oracle $\mathcal{O}$ is defined below, and the bit α is defined as follows: if $L_0 = L_1$, then $\alpha = \beta$, otherwise, $\alpha = 0$.

- *$\mathcal{O}(v_0, v_1)$ executes $L_0 \leftarrow L_0 \cup \{v_0\}; L_1 \leftarrow L_1 \cup \{v_1\}; b_0 \leftarrow \mathsf{Vote}_{pk}(v_0); b_1 \leftarrow \mathsf{Vote}_{pk}(v_1); \mathfrak{bb}_0 \leftarrow \mathsf{BB}(\mathfrak{bb}_0, b_0); \mathfrak{bb}_1 \leftarrow \mathsf{BB}(\mathfrak{bb}_1, b_1)$, if $v_0, v_1 \in \mathfrak{m}$.*
- *$\mathcal{O}(b)$ assigns $\mathfrak{bb}'_\beta \leftarrow \mathfrak{bb}_\beta$, executes $\mathfrak{bb}_\beta \leftarrow \mathsf{BB}(\mathfrak{bb}_\beta, b)$ and if $\mathfrak{bb}_\beta \neq \mathfrak{bb}'_\beta$, then executes $\mathfrak{bb}_{1-\beta} \leftarrow \mathsf{BB}(\mathfrak{bb}_{1-\beta}, b)$.*
- *$\mathcal{O}()$ outputs $\mathfrak{bb}_\beta$.*

We say Γ satisfies ballot secrecy *if for all probabilistic polynomial-time adversaries $\mathcal{A}$ and security parameters n, there exists a negligible function* negl *such that IND-SEC$_{\mathcal{A},\Gamma}(n) \leq \mathsf{negl}(n)$.*

Our game captures a setting where an adversary can cast ballots on behalf of a subset of voters, whom we call dishonest voters, and controls the distribution of votes cast by the remaining voters, whom we call honest voters, but honest voters always cast ballots constructed by the Vote algorithm. Furthermore, at the end of the election, the adversary obtains the election result. Intuitively, if the adversary loses the game, then the adversary is unable to distinguish between the bulletin boards $\mathfrak{bb}_0$ and $\mathfrak{bb}_1$, hence, the adversary cannot distinguish between an honest ballot $b_0 \in \mathfrak{bb}_0$ and an honest ballot $b_1 \in \mathfrak{bb}_1$, therefore, voters' votes cannot be revealed. On the other hand, if the adversary wins the game, then there exists a strategy to distinguish honestly cast ballots. We stress that a unanimous election result will always reveal all voters' votes and we tolerate this factor in our game by challenging the adversary to guess the bit β, rather than the distribution of votes.

Comparing IND-SEC and earlier definitions. In comparison with earlier definitions by Bernhard *et al.* [6–8], Definition 5 permits $\alpha \in \{0, 1\}$, whereas, earlier presentations implicitly[4] insist $\alpha = 0$. It follows that Definition 5 allows the adversary to access auxiliary data generated by tallying $\mathfrak{bb}_\beta$, whereas, earlier definitions only allow the adversary to access the auxiliary data generated by tallying $\mathfrak{bb}_0$. Accordingly, earlier definitions implicitly assume that auxiliary data cannot be used to violate ballot secrecy, indeed, this corresponds to the description by Bernhard *et al.* [6, §2.2]: *"[ballot secrecy] is satisfied if an adversary [...] cannot learn anything about the votes of [...] honest voters beyond what can be inferred from the election result."* Unfortunately, however, it is possible that the auxiliary data can reveal voters' votes. For example, a variant of Enc2Vote (Definition 4) could define auxiliary data that maps ballots to decrypted ballots, thereby violating ballot secrecy; indeed, as highlighted in Footnote 3, Bernhard, Pereira & Warinschi [7] provided such a mapping in their variant of Enc2Vote. As discussed, we permit $\alpha \in \{0, 1\}$, rather than $\alpha = 0$, thereby strengthening Definition 5 in comparison with earlier definitions and, thus, overcoming the limitations of previous works.

4 Ballot Independence

Intuitively, if an election scheme satisfies ballot independence, then an adversary is unable to construct a ballot that will be accepted by the election's bulletin board *and* be meaningfully related to a non-adversarial ballot from the bulletin board [4, Section 7.2], thereby capturing the notion that accepted ballots are tamper-proof. Building upon inspiration from non-malleable encryption, we formalise ballot independence as a non-malleability game.

4.1 Non-malleability Game

The concept of non-malleability and first formalisation is due to Dolev, Dwork & Naor [18,20]. Bellare *et al.* [19] build upon these results to introduce NM-CPA

[4] Earlier presentations do not explicitly define a bit α, however, they always tally $\mathfrak{bb}_0$ and this implicitly corresponds to $\alpha = 0$ in Definition 5.

(Definition 2) and based upon NM-CPA, we formalise ballot independence (Definition 6) as a pair of cryptographic games: $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{A},\Pi}$ and $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{A},\Pi,\$}$. The first three steps of both games are identical. First, the challenger sets up the keys, vote space, and bulletin board. Secondly, the adversary gets the vote space $\mathfrak{m}$, the public key pk and the board $\mathfrak{bb}$ as input and must return a distribution M on the vote space. The adversary may also read the board and submit ballots of his own. Thirdly, the challenger samples a vote v from M. At this point the two games diverge: in $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{A},\Pi}$, the challenger constructs a ballot $\mathsf{Vote}_{pk}(v)$ and adds it to the bulletin board; whereas, in $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{A},\Pi,\$}$, the challenger samples a second vote v' from M, constructs a ballot $\mathsf{Vote}_{pk}(v')$ and adds it to the bulletin board. Fourthly, the adversary must compute a relation R which is intended to distinguish the election results produced by the two games. Finally, the challenger tallies the election and evaluates the relation R on the vote v and, after removing the challenge vote, the election result. The adversary's advantage is the difference between the probabilities that his relation is satisfied in each game.

Definition 6 (NM-BB: Ballot independence). *Let $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ be an election scheme, $\mathcal{A} = (A_1, A_2)$ be an adversary, and*

$$\mathit{NM\text{-}BB}_{\mathcal{A},\Gamma}(n) := |\mathit{Succ}^{BB}_{\mathcal{A},\Pi}(n) - \mathit{Succ}^{BB}_{\mathcal{A},\Pi,\$}(n)|$$

where $\mathit{Succ}^{BB}_{\mathcal{A},\Pi}(n)$ and $\mathit{Succ}^{BB}_{\mathcal{A},\Pi,\$}(n)$ are defined below, and n is the security parameter.

$$\begin{aligned}\mathit{Succ}^{BB}_{\mathcal{A},\Pi}(n) = Pr[&(\mathfrak{bb}, \mathfrak{m}, pk, sk) \leftarrow \mathsf{Setup}(1^n);\ (M, s) \leftarrow A_1^{\mathcal{O}}(\mathfrak{m}, pk);\\ &v \leftarrow M;\ b \leftarrow \mathsf{Vote}_{pk}(v);\ \mathfrak{bb} \leftarrow \mathsf{BB}(\mathfrak{bb}, b);\ R \leftarrow A_2^{\mathcal{O}}(s);\\ &(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}) : R(v, \mathfrak{v}\backslash\{v\})]\end{aligned}$$

$$\begin{aligned}\mathit{Succ}^{BB}_{\mathcal{A},\Pi,\$}(n) = Pr[&(\mathfrak{bb}, \mathfrak{m}, pk, sk) \leftarrow \mathsf{Setup}(1^n);\ (M, s) \leftarrow A_1^{\mathcal{O}}(\mathfrak{m}, pk);\\ &v, v' \leftarrow M;\ b \leftarrow \mathsf{Vote}_{pk}(v');\ \mathfrak{bb} \leftarrow \mathsf{BB}(\mathfrak{bb}, b);\ R \leftarrow A_2^{\mathcal{O}}(s);\\ &(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}) : R(v, \mathfrak{v}\backslash\{v'\})]\end{aligned}$$

In the above games we let $\mathcal{O}$ be defined as follows: $\mathcal{O}(b)$ executes $\mathfrak{bb} \leftarrow \mathsf{BB}(\mathfrak{bb}, b)$ and $\mathcal{O}()$ outputs $\mathfrak{bb}$. Moreover, we insist the vote space sampling algorithm M and the relation R are computable in polynomial time, and for all $v \leftarrow M$ we have $v \in \mathfrak{m}$. We say Γ satisfies NM-BB (or ballot independence*) if for all probabilistic polynomial-time adversaries $\mathcal{A}$ and security parameters n, there exists a negligible function* negl *such that $\mathit{NM\text{-}BB}_{\mathcal{A},\Gamma}(n) \leq \mathsf{negl}(n)$.*

Intuitively, if an adversary wins the game, then the adversary is able to construct a relation R which holds for a challenge ballot $b \leftarrow \mathsf{Vote}_{pk}(v)$ but fails for $b \leftarrow \mathsf{Vote}_{pk}(v')$. However, we must avoid crediting the adversary for trivial and unavoidable relations which hold iff the challenge vote appears in the election result, hence, we remove the challenge vote from the election result. By contrast, if the adversary can derive a ballot containing the challenge vote and the bulletin board accepts such a ballot, then the adversary can win the game. For example, suppose an election scheme allows the bulletin board to accept duplicate ballots and witness that an adversary can win the game as follows, namely, the adversary selects M as a uniform distribution on $\mathfrak{m}$, calls $\mathcal{O}(b)$ with the challenge ballot b, and defines a relation $R(v, \mathfrak{v})$ that holds iff $v \in \mathfrak{v}$. In this setting, $R(v, \{v\})$ always holds at the end of $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{A},\Pi}$, whereas, $R(v, \{v'\})$ holds with probability $1/\mathfrak{m}$ at the end of $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{A},\Pi,\$}$, since v' is sampled independently from v. Finally, if an adversary loses the game, then the adversary is unable to construct a suitable relation, hence, there is no ballot which the bulletin board will accept such that the ballot is related to $\mathsf{Vote}_{pk}(v)$ but not $\mathsf{Vote}_{pk}(v')$, therefore, the adversary cannot cast a ballot which is meaningfully related to an honest voter's ballot.

Comparing NM-BB and NM-CPA. The main distinction between the notion of non-malleability (Definition 2) and our definition of ballot independence is: NM-CPA universally quantifies over ciphertexts, whereas, NM-BB quantifies over ballots accepted by the bulletin board. It follows that non-malleability for encryption is intuitively stronger than ballot independence, since non-malleability for encryption insists that the adversary cannot construct ciphertexts meaningfully related to the challenge ciphertext, whereas, ballot independence tolerates meaningfully related ballots, assuming that they are rejected by the bulletin board algorithm BB. For example, suppose an adversary $\mathcal{A}$ includes the challenge ciphertext in the vector $\mathbf{y}$ and observe that this adversary cannot win $\mathsf{NM\text{-}CPA}_{\mathcal{A},\Pi}(n)$, due to the constraint $y \notin \mathbf{y}$; by comparison, suppose an adversary $\mathcal{B}$ copies the challenge ballot b and observe that this adversary can win $\mathsf{NM\text{-}BB}_{\mathcal{B},\Gamma}(n)$. Nonetheless, for ballot independence, the bulletin board must not contain meaningfully related ballots and, hence, checking for meaningfully related ballots is a prerequisite of the bulletin board algorithm BB.

Non-malleable Ballots are Sufficient. Non-malleability for encryption prevents the adversary from constructing a ciphertext meaningfully related to the challenge ciphertext and, hence, it follows that non-malleable ballots are sufficient for ballot independence. Indeed, we can derive non-malleable ballots in our Enc2Vote construction using encryption schemes satisfying NM-CPA.

Proposition 7. *Given an encryption scheme Π satisfying NM-CPA, the election scheme* $\mathsf{Enc2Vote}(\Pi)$ *satisfies ballot independence.*

In Proposition 7, it is sufficient for the bulletin board algorithm, defined by $\mathsf{Enc2Vote}(\Pi)$, to reject ballots that already appear on the bulletin board since non-malleability prevents the adversary from creating ballots meaningfully related to honest voters' votes (except for exact copies). The proof is essentially the same as that of [7, Theorem 4.2].

4.2 Indistinguishability Game

Our non-malleability game (NM-BB) captures an intuitive notion of ballot independence, however, the definition is relatively complex and security proofs in this setting are relatively difficult. Bellare & Sahai [21] observed similar complexities with definitions of non-malleability for encryption and show that NM-CPA is equivalent to a simpler, indistinguishability-based notion. In a similar direction, we introduce an indisinguishability game IND-BB for ballot independence and, based upon Bellare & Sahai's proof, show that our games NM-BB and IND-BB are equivalent.

We model ballot independence as an indistinguishability game between an adversary and a challenger (Definition 8). Informally, the game proceeds as follows. First, the challenger initialises the bulletin board $\mathfrak{bb}$, defines the vote space $\mathfrak{m}$, and constructs a key pair (pk, sk). Secondly, the adversary executes the algorithm A_1 on the public key pk and vote space $\mathfrak{m}$, and outputs the triple (v_0, v_1, s), where $v_0, v_1 \in \mathfrak{m}$ and s is some state information. Thirdly, the challenger randomly selects a bit β, computes a challenge ballot b, and updates the bulletin board with b. Fourthly, the adversary executes the algorithm A_2 which outputs some state t. Next, the challenger computes the election result $\mathfrak{v}$. Finally, the adversary executes the algorithm A_3 on the input t and $\mathfrak{v}\backslash\{v_\beta\}$. The election scheme satisfies ballot independence if the adversary has less than a negligible advantage over guessing the bit β.

Definition 8 (IND-BB: Ballot independence). *Let Γ = (Setup, Vote, BB, Tally) be an election scheme, $\mathcal{A} = (A_1, A_2, A_3)$ be an adversary, n be the security parameter and IND-BB$_{\mathcal{A},\Gamma}(n)$ the cryptographic game defined below.*

$$2 \cdot Pr[(\mathfrak{bb}, \mathfrak{m}, pk, sk) \leftarrow \mathsf{Setup}(1^n);\ (v_0, v_1, s) \leftarrow A_1^{\mathcal{O}}(\mathfrak{m}, pk);\ \beta \leftarrow \{0,1\};$$
$$b \leftarrow \mathsf{Vote}_{pk}(v_\beta);\ \mathfrak{bb} \leftarrow \mathsf{BB}(\mathfrak{bb}, b);\ t \leftarrow A_2^{\mathcal{O}}(s);\ (\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}) :$$
$$A_3(t, \mathfrak{v}\backslash\{v_\beta\}) = \beta] - 1$$

In the above game we let $\mathcal{O}$ be defined as follows:

- $\mathcal{O}(b)$ *executes* $\mathfrak{bb} \leftarrow \mathsf{BB}(\mathfrak{bb}, b)$
- $\mathcal{O}()$ *outputs* $\mathfrak{bb}$

Moreover, we insist that $v_0, v_1 \in \mathfrak{m}$. We say Γ satisfies IND-BB (or ballot independence*) if for all probabilistic polynomial-time adversaries $\mathcal{A}$ and security parameters n, there exists a negligible function* negl *such that IND-BB$_{\mathcal{A},\Gamma}(n) \leq$* $\mathsf{negl}(n)$.

Intuitively, if an adversary wins the game, then the adversary is able to distinguish between challenge ballots $b \leftarrow \mathsf{Vote}_{pk}(v_0)$ and $b \leftarrow \mathsf{Vote}_{pk}(v_1)$. As per our NM-BB game, we avoid trivial and unavoidable distinctions by removing the challenge vote from the election result.

Our ballot independence games are based on standard security models for encryption: NM-BB is based on non-malleability whereas IND-BB game is based on indistinguishability. Bellare and Sahai [21] have shown that non-malleability is equivalent to a notion of indistinguishability for encryption and we adapt their proof to show that $\mathsf{NM\text{-}BB}$ and $\mathsf{IND\text{-}BB}$ are equivalent.

Theorem 9 ($\mathsf{NM\text{-}BB} = \mathsf{IND\text{-}BB}$). *Given an election scheme Γ, we have Γ satisfies $\mathsf{NM\text{-}BB}$ if and only if Γ satisfies $\mathsf{IND\text{-}BB}$.*

Theorem 9 relates the advantage of an adversary casting a vote meaningfully related to an honest voter's vote to an advantage in guessing the honest voter's vote, in a setting where the election result does not contain the honest voter's vote. The proof of Theorem 9 can be found in the full version of our paper [1].

4.3 Controlled Malleability Is Sufficient

Recall that ballot independence tolerates meaningfully related ballots, assuming they are rejected by the bulletin board. It follows intuitively that we can weaken the requirement for an $\mathsf{NM\text{-}CPA}$ encryption scheme in Proposition 7, assuming we modify $\mathsf{Enc2Vote}$'s bulletin board algorithm to reject ballots meaningfully related to existing ballots on the bulletin board. We start with a simple example. Given an encryption scheme satisfying $\mathsf{NM\text{-}CPA}$, we can derive a new encryption scheme by prepending a random bit to all ciphertexts and removing this bit before decryption. This new encryption scheme does not satisfy $\mathsf{NM\text{-}CPA}$, however, we can derive an election scheme satisfying ballot independence using $\mathsf{Enc2Vote}$ if we modify $\mathsf{Enc2Vote}$'s bulletin board algorithm as follows: given a bulletin board $\mathfrak{bb}$ and ballot b, reject b if it is identical to any ballot already on $\mathfrak{bb}$ up to the first bit. This example shows that non-malleable ballots are not necessary for ballot independence. Let us now formalise a notion of *controlled malleability*[5], denoted $\mathsf{NM\text{-}CPA}/R$ (pronounced "NM-CPA modulo R"), which we will show is sufficient for ballot independence.

Definition 10 (Controlled malleability). *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an asymmetric encryption scheme and R be an efficiently computable equivalence relation on Π's ciphertext space. We say that Π satisfies $\mathsf{NM\text{-}CPA}/R$ (or* controlled malleability*) if for all efficient adversaries $\mathcal{A}$ the following probability is negligible*

$$Pr\left[(pk, sk) \leftarrow \mathsf{Gen}(1^n); \beta \leftarrow \{0,1\} \ : \ \mathcal{A}^{\mathsf{chal}_\beta, \mathsf{dec}}(pk) = \beta\right]$$

where the oracles chal *and* dec *are defined as follows and each oracle may be called once, in any order.*

- chal_β *takes two messages m_0 and m_1 of equal length as input, computes $c^* \leftarrow \mathsf{Enc}_{pk}(m_\beta)$, and outputs c^*.*
- dec *takes a vector $\mathbf{c}$ of ciphertexts as input. If chal_β has previously output a ciphertext c^* such that $R(c, c^*)$ holds for some $c \in \mathbf{c}$, then output $\perp$, otherwise, output $\mathsf{Dec}_{sk}(\mathbf{c})$.*

[5] The term is taken from Kohlweiss et al. [22] who introduce controlled malleability for zero-knowledge proofs.

Our definition generalises non-malleability for encryption, in particular, $\mathsf{NM\text{-}CPA} = \mathsf{NM\text{-}CPA}/R$, when R is the identity. Moreover, we note that our definition could be adapted to a notion of $\mathsf{CCA2}/R$ by allowing arbitrarily many decryption queries. The construction $\mathsf{Enc2Vote}$ can be generalised to asymmetric encryption schemes satisfying controlled malleability as follows.

Definition 11 ($\mathsf{Enc2Vote}/R$). *Suppose* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is an asymmetric encryption scheme and R is an efficiently computable equivalence relation on Π's ciphertext space, we define* $\mathsf{Enc2Vote}/R(\Pi) = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ *as follows. Let the* Setup*,* Vote *and* Tally *algorithms be given by* $\mathsf{Enc2Vote}(\Pi)$*. The* BB *algorithm takes $\mathfrak{bb}$ and b as input, where $\mathfrak{bb}$ is a multiset. If there exists $b' \in \mathfrak{bb}$ such that $R(b, b')$, then* BB *outputs $\mathfrak{bb}$, otherwise,* BB *outputs $\mathfrak{bb} \cup \{b\}$.*

Assuming that the relation R does not relate fresh, honestly generated ciphertexts in Π's ciphertext space to other values (Definition 12), we can ensure that $\mathsf{Enc2Vote}/R(\Pi)$ satisfies the correctness condition of election schemes and, hence, $\mathsf{Enc2Vote}/R(\Pi)$ is an election scheme satisfying ballot independence by (Proposition 13).

Definition 12 (Sparse relation). *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an asymmetric encryption scheme and R be an efficiently computable equivalence relation on Π's ciphertext space. We say R is a* sparse relation *if for all $(pk, sk) \leftarrow \mathsf{Gen}$, c and m, we have $c' \leftarrow \mathsf{Enc}(m, pk)$ yields $R(c, c') = 0$ with overwhelming probability.*

Proposition 13. *Suppose Π is an asymmetric encryption scheme and R is an efficiently computable and sparse equivalence relation on Π's ciphertext space such that Π satisfies NM-CPA/R. We have* $\mathsf{Enc2Vote}/R(\Pi)$ *satisfies ballot independence.*

The proof of Proposition 13 is similar to the proof of [7, Theorem 4.2].

Design Paradigms and Discussion. We derive the following design paradigms from our results: 1) use non-malleable ballots (Section 4.1), or 2) identify and reject related ballots using controlled malleability. The latter paradigm is particularly useful when ballots contain malleable data such as voter identities or pseudonyms, since we can tolerate malleability and provide provable security. Moreover, it facilitates more realistic models of election schemes in comparison with earlier work, for example, Bernhard *et al.* [6–8] abstractly model Helios ballots as non-malleable ciphertexts, whereas, in practice, Helios ballots embed non-malleable ciphertexts in malleable JavaScript Object Notation (JSON) data structures (this is particularly relevant, since Smyth & Cortier [23, §4.1] have shown that the JSON structures introduces vulnerabilities).

5 Ballot Secrecy Implies Ballot Independence

In this paper, all election schemes satisfy correctness: the bulletin board algorithm BB adds honestly constructed ballots to the bulletin board, the tally algorithm Tally includes honest votes in the election result, and the number of votes in an election result corresponds to the number of ballots (that is, each ballot contains one vote). In this setting, an election scheme satisfying ballot secrecy also satisfies ballot independence.

Theorem 14 (Ballot secrecy implies ballot independence). *Given an election scheme Γ satisfying ballot secrecy, we have Γ satisfies ballot independence.*

Proof (Proof sketch). The proof is by a standard reduction argument: given a successful $\mathsf{IND\text{-}BB}$ adversary, we construct an adversary against $\mathsf{IND\text{-}SEC}$. The single challenge query on (v_0, v_1) becomes a pair of vote queries $\mathsf{Vote}(v_0, v_1)$ and $\mathsf{Vote}(v_1, v_0)$, and oracle queries $\mathcal{O}(b)$ become ballot queries. When we obtain the election outcome from the $\mathsf{IND\text{-}SEC}$ game, we remove v_0 and v_1 since this is the distribution that the $\mathsf{IND\text{-}BB}$ adversary expects. Finally, we show that the advantage translates between games. □

Theorem 14 relates an advantage in guessing an honest voter's vote in a setting where the election result *does not* contain the honest voter's vote to an advantage in the ballot secrecy game where the election result *does* include the honest voter's vote. It follows, by Theorem 9, that an advantage in casting a vote meaningfully related to an honest voter's vote translates into an advantage in guessing an honest voter's vote, hence, we have shown that ballot independence is necessary for ballot secrecy in election schemes defined by Definition 3. The proof of Theorem 14 can be found in the full version of our paper [1].

5.1 Critique of Desmedt and Chaidos's Helios Variant

Intuitively, Theorem 14 contradicts the results by Desmedt & Chaidos [16], who claim to provide a variant of the Helios election scheme which allows Bob to cast the same vote as Alice, with Alice's cooperation, whilst preventing Bob from learning Alice's vote. In their protocol, Bob selects Alice's ballot from the bulletin board and communicates with Alice to generate a new ballot that is guaranteed to contain the same vote as Alice's. Desmedt & Chaidos's security claim is true *before the election result is announced*, since Bob gains no advantage in guessing Alice's vote. However, *after the election result is announced*, the claim is false. We can informally contradict this claim – using results by Cortier & Smyth [4,10,11] – in an election with voters Alice, Bob and Charlie: if Bob casts the same vote as Alice, then Bob can learn Alice's vote by observing the election result and checking which candidate obtained at least two votes (that is, Bob can learn Alice's vote when the election result is not unanimous). We believe the erroneous claim by Desmedt & Chaidos is due to an invalid inference from their computational security result. Indeed, although the result [16, Theorem 1] is correct, their model does not support their claims for real world security: Desmedt & Chaidos consider a passive adversary that cannot observe the election result, whereas, we believe a practical notion of security must consider an *active* adversary who can cast ballots and observe the election result, since this captures the capabilities of an attacker in the real world. Nonetheless, a weaker notion of ballot secrecy may be satisfiable in Desmedt & Chaidos's variant of Helios, assuming Alice never cooperates with the adversary. Clearly, no claims can be made about Bob's knowledge of Alice's vote in this setting. We have shown Desmedt & Chaidos our results and Chaidos agrees with our findings [24].

5.2 Discussion

We have shown that election schemes satisfying ballot secrecy must also satisfy ballot independence. However, we must concede that alternative formalisms of election schemes may permit different results. Indeed, Cortier & Smyth [4, Section 7.1] present a result to the contrary using anonymous channels, which are implicitly excluded from our model. Moreover, our model also excludes settings where the adversary cannot control a majority of voters and places some restrictions on the election result, namely, the election result is captured as a multiset which reveals the number of votes for each candidate. In this setting, an election result can be computed from a partial election result if the votes of the remaining voters are known. This property is implicitly used in our proof of Theorem 14. On the other hand, some practical election schemes do not have this property. For example, consider an election scheme which announces the winning candidate, but does not provide a breakdown of the votes for each candidate [25–28]. It follows that knowledge of a partial election result can only be used to derive the election result if the adversary controls a majority of voters. Similarly, given an election result and knowledge of a minority of votes, a partial election result which excludes the known votes cannot be derived. In this setting, we believe election schemes can satisfy ballot secrecy but not ballot independence, since casting a minority of related ballots is not sufficient to reveal a voter's vote. Formal treatment of this case and consideration of whether such schemes are practical is a possible direction for future work.

6 Sufficient Conditions for Ballot Secrecy

The main distinctions between our ballot secrecy ($\mathsf{IND\text{-}SEC}$) and ballot independence ($\mathsf{IND\text{-}BB}$) games are as follows.

1. The challenger in our ballot independence game explicitly defines a challenge ballot and adds the ballot to the bulletin board, whereas, the challenger in our ballot secrecy game provides the adversary with an oracle $\mathcal{O}_{\mathcal{B}}(\cdot, \cdot)$.

The two formulations are similar, indeed, the challenger's computation $b \leftarrow \mathsf{Vote}_{pk}(v_\beta)$; $\mathfrak{bb} \leftarrow \mathsf{BB}(\mathfrak{bb}, b)$ is similar to an oracle call $\mathcal{O}_{\mathcal{B}}(v_0, v_1)$. Moreover, a hybrid argument will show that it does not matter if we give the adversary only one challenge ballot or many oracle calls.

2. The adversary in our ballot secrecy game has access to the auxiliary data produced during tallying, but the adversary in our ballot independence game does not.

The second point distinguishes our two games; Theorem 14 shows that ballot secrecy is stronger than independence and Footnote 3 gives a case where it is strictly stronger: the presentation of the $\mathsf{Enc2Vote}$ construction by Bernhard, Pereira & Warinschi provides ballot independence, but the auxiliary data maps voters to votes, thereby violating ballot secrecy. Nonetheless, by restricting the adversary's access to auxiliary data we can show that the two games are equivalent (Theorem 15) and, hence, in the absence of auxiliary data, ballot independence is a sufficient condition for ballot secrecy, in particular, $\mathsf{Enc2Vote}$ and $\mathsf{Enc2Vote}/R$ are constructions for election schemes satisfying ballot secrecy.

Theorem 15 ($\mathsf{NM\text{-}BB} = \mathsf{IND\text{-}SEC}$, without auxiliary data). *Suppose $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ is an election scheme such that there exists a constant symbol $\perp$ and for all parameters $(\mathfrak{bb}_0, \mathfrak{m}, pk, sk) \leftarrow \mathsf{Setup}(1^n)$, multiesets $\mathfrak{bb}$ and tallying data $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb})$, we have $aux = \perp$. It follows that Γ satisfies ballot secrecy if and only if Γ satisfies ballot independence.*

A proof of Theorem 15 can be found in the full version of this paper [1]. In essence, the proof uses a standard hybrid argument to show that it is sufficient to consider a variant of the $\mathsf{IND\text{-}SEC}$ game in which the adversary is restricted to a single oracle call $\mathcal{O}(v_0, v_1)$ and shows that an adversary in this game can be used to construct a successful adversary against $\mathsf{IND\text{-}BB}$.

Intuitively, we can generalise Theorem 15 to election schemes in which the auxiliary data can be simulated. Since the auxiliary data output by election schemes typically consists of signatures of knowledge proving that the election result has been correctly computed from the bulletin board, we expect many practical election schemes will satisfy zero-knowledge auxiliary data, indeed, Helios outputs partial ElGamal decryptions [29, 30] and proofs demonstrating knowledge of discrete logarithms [31–33] which can be simulated. In this context, we believe ballot secrecy and ballot independence coincide (Remark 16). Unfortunately, formalising zero-knowledge is a complex issue – in particular, the simulator needs some extra capabilities compared to the election officials (otherwise the officials could publish simulated proofs!) – to which there is no general solution and, hence, there is no general proof of Remark 16. Nonetheless, we believe Remark 16 can be shown to hold for particular formalisations of zero-knowledge, for instance, a proof could be constructed in the programmable random oracle model (the proof would essentially be that of Theorem 15 with the simulator being run at the appropriate point; we briefly comment on this in the proof of Theorem 15) and, hence, a proof of ballot secrecy can be reduced to a proof of ballot independence.

Remark 16 ($\mathsf{NM\text{-}BB} = \mathsf{IND\text{-}SEC}$ for zero-knowledge auxiliary data). Given an election scheme Γ satisfying zero-knowledge auxiliary data (informally, zero-knowledge auxiliary data means that the auxiliary data can be simulated given the result), we have Γ satisfies ballot secrecy if and only if Γ satisfies ballot independence.

Remark 16 suggests that ballot independence is a sufficient condition for ballot secrecy in election schemes where auxiliary data can be simulated. Coupled with earlier results [8], this should facilitate a proof of ballot secrecy in Helios. (Bernhard *et al.* [6] provide a proof of ballot secrecy in a variant of Helios which uses the Naor & Yung transformation [34] to derive non-malleable ballots and Bernhard, Pereira & Warinschi [8] prove that Helios satisfies ballot secrecy in the special case of referendums, however, a full proof of ballot secrecy in Helios is not currently known.)

7 Conclusion

We have formalised *ballot independence* in a variant of the model for election schemes proposed by Bernhard *et al.* Our main results are as follows. Ballot secrecy implies ballot independence; the converse holds too if there is no auxiliary data. Moreover, we have argued that ballot independence and ballot secrecy coincide if auxiliary data is "zero knowledge;" since auxiliary data typically consists of zero knowledge proofs, this assumption is realistic and holds for election schemes such as Helios, for instance. Furthermore, we provide some sufficient conditions for ballot independence and, hence, ballot secrecy: we show that non-malleable ballots are sufficient but not necessary for independence and secrecy, and introduce a weaker notion of controlled-malleable encryption which we show is sufficient, moreover, this notion is better suited to modelling the way ballots are handled in practice (for example, by Helios). In addition, we show that the notion of ballot secrecy proposed by Bernhard *et al.* does not capture attacks which rely on auxiliary data and we adopt a stronger definition. Furthermore, we show that the variant of Helios proposed by Desmedt & Chaidos does not satisfy ballot secrecy.

Acknowledgements. We are particularly grateful to Bogdan Warinschi and the anonymous reviewers who read earlier versions of this paper and provided useful guidance. This work has been partly supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC project *CRYSP* (259639) and by EPSRC via grant EP/H043454/1.

References

1. Smyth, B., Bernhard, D.: Ballot secrecy and ballot independence coincide. Cryptology ePrint Archive, Report 2013/235 (2013)
2. Delaune, S., Kremer, S., Ryan, M.: Coercion-Resistance and Receipt-Freeness in Electronic Voting. In: CSFW 2006: 19th Computer Security Foundations Workshop, pp. 28–42. IEEE Computer Society (2006)
3. Backes, M., Hriţcu, C., Maffei, M.: Automated Verification of Remote Electronic Voting Protocols in the Applied Pi-calculus. In: CSF 2008: 21st Computer Security Foundations Symposium, pp. 195–209. IEEE Computer Society (2008)
4. Cortier, V., Smyth, B.: Attacking and fixing Helios: An analysis of ballot secrecy. Journal of Computer Security 21(1), 89–148 (2013)

5. Delaune, S., Kremer, S., Ryan, M.D.: Verifying privacy-type properties of electronic voting protocols. Journal of Computer Security 17(4), 435–487 (2009)
6. Bernhard, D., Cortier, V., Pereira, O., Smyth, B., Warinschi, B.: Adapting Helios for provable ballot privacy. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 335–354. Springer, Heidelberg (2011)
7. Bernhard, D., Pereira, O., Warinschi, B.: On Necessary and Sufficient Conditions for Private Ballot Submission. Cryptology ePrint Archive, Report 2012/236 (version 20120430:154117b) (2012)
8. Bernhard, D., Pereira, O., Warinschi, B.: How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 626–643. Springer, Heidelberg (2012)
9. Gennaro, R.: Achieving independence efficiently and securely. In: PODC 1995: 14th Principles of Distributed Computing Symposium, pp. 130–136. ACM Press (1995)
10. Smyth, B., Cortier, V.: 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/
11. Cortier, V., Smyth, B.: Attacking and fixing Helios: An analysis of ballot secrecy. In: CSF 2011: 24th Computer Security Foundations Symposium, pp. 297–311. IEEE Computer Society (2011)
12. Schneier, B.: Hacking the Papal Election (2013), https://www.schneier.com/blog/archives/2013/02/hacking_the_pap.html
13. Bulens, P., Giry, D., Pereira, O.: Running Mixnet-Based Elections with Helios. In: EVT/WOTE 2011: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections. USENIX Association (2011)
14. Adida, B.: Helios: Web-based Open-Audit Voting. In: USENIX Security 2008: 17th USENIX Security Symposium, pp. 335–348. USENIX Association (2008)
15. Adida, B., Marneffe, O., Pereira, O., Quisquater, J.: Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios. In: EVT/WOTE 2009: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections. USENIX Association (2009)
16. Desmedt, Y., Chaidos, P.: Applying Divertibility to Blind Ballot Copying in the Helios Internet Voting System. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012. LNCS, vol. 7459, pp. 433–450. Springer, Heidelberg (2012)
17. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults. In: FOCS 1985: 26th Foundations of Computer Science Symposium, pp. 383–395. IEEE Computer Society (1985)
18. Dolev, D., Dwork, C., Naor, M.: Non-Malleable Cryptography. In: STOC 1991: 23rd Theory of Computing Symposium, pp. 542–552. ACM Press (1991)
19. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations Among Notions of Security for Public-Key Encryption Schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998)
20. Dolev, D., Dwork, C., Naor, M.: Nonmalleable Cryptography. Journal on Computing 30(2), 391–437 (2000)
21. Bellare, M., Sahai, A.: Non-malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 519–536. Springer, Heidelberg (1999)
22. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Malleable Proof Systems and Applications. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 281–300. Springer, Heidelberg (2012)
23. Smyth, B., Cortier, V.: Does Helios ensure ballot secrecy? Cryptology ePrint Archive, Report 2010/625 (version 20101217:132825) (2010)

24. Chaidos, P.: Private email communication (March/April 2013)
25. Benaloh, J., Yung, M.: Distributing the Power of a Government to Enhance the Privacy of Voters. In: PODC 1986: 5th Principles of Distributed Computing Symposium, pp. 52–62. ACM Press (1986)
26. Hevia, A., Kiwi, M.: Electronic Jury Voting Protocols. In: Rajsbaum, S. (ed.) LATIN 2002. LNCS, vol. 2286, pp. 415–429. Springer, Heidelberg (2002)
27. Hevia, A., Kiwi, M.A.: Electronic jury voting protocols. Theoretical Computer Science 321(1), 73–94 (2004)
28. Desmedt, Y., Kurosawa, K.: Electronic Voting: Starting Over? In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 329–343. Springer, Heidelberg (2005)
29. Pedersen, T.P.: A Threshold Cryptosystem without a Trusted Party. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 522–526. Springer, Heidelberg (1991)
30. Chaum, D., Pedersen, T.P.: Wallet Databases with Observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993)
31. Chaum, D., Evertse, J.-H., van de Graaf, J., Peralta, R.: Demonstrating Possession of a Discrete Logarithm Without Revealing It. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 200–212. Springer, Heidelberg (1987)
32. Chaum, D., Evertse, J.-H., van de Graaf, J.: An Improved Protocol for Demonstrating Possession of Discrete Logarithms and Some Generalizations. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 127–141. Springer, Heidelberg (1988)
33. Schnorr, C.-P.: Efficient Identification and Signatures for Smart Cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (1990)
34. Naor, M., Yung, M.: Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In: STOC 1990: 22nd Theory of Computing Symposium, pp. 427–437. ACM Press (1990)