Ballot secrecy and ballot independence: definitions and relations

Ben Smyth[1] and David Bernhard[2]

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

October 9, 2014

Abstract

We study ballot independence for election schemes. First, we formally define ballot independence as a cryptographic game and prove that ballot secrecy implies ballot independence. Secondly, we introduce a notion of controlled malleability and prove that it is sufficient for ballot independence. We also prove that non-malleable ballots are sufficient for ballot independence. Thirdly, we prove that ballot independence is sufficient for ballot secrecy in a special case. 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] [DKR06,BHM08,CS13] 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* [DKR09] provide privacy in environments where intimidation may occur. Bernhard *et al.* [BCP+11, BPW12b, BPW12a] propose a cryptographic formalisation of ballot secrecy.

*An earlier version [SB13] of this paper was presented at ESORICS'13.

[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.

However, their definition does not capture the publication of tallying proofs[2] and we extend the definition of ballot secrecy by Bernhard *et al.* to support the publication of such proofs[3] (Section 3).

Ballot independence [Gen95, CS13] 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.

Indeed, Cortier & Smyth [CS13, SC11, CS11] 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. In Section 4, we propose a definition of ballot independence and give sufficient conditions to achieve this notion, including a construction for election schemes from encryption schemes satisfying our notion of *controlled-malleable encryption* (a generalisatin of non-malleable encryption).

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 [Sch13] 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 [BGP11, §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 [Adi08, AMPQ09] election scheme, Desmedt & Chaidos [DC12] 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. We prove that ballot secrecy implies ballot independence (Section 5), 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 the results by Desmedt & Chaidos and argue that their security results do not support their claims.

Finally, we present a class of election schemes for which ballot secrecy and ballot independence coincide (Section 6). It follows that our construction for

[2] The ESORICS'13 version of this paper [SB13] incorrectly claims that the definition of ballot secrecy by Bernhard *et al.* [BCP$^+$11, BPW12b, BPW12a] allows election schemes that reveal voters' votes to be proven secure. This erroneous claim was made on the basis that definitions by Bernhard *et al.* gave the adversary access to tallying proofs, which appears to be true with reference to [BCP$^+$11, Algorithm 4], but is forbidden by the correctness property [BCP$^+$11, Figure 1].

[3] Galindo & Cortier [GC13] have shown that our original presentation of ballot secrecy [SB13, Definition 5] is too strong, since it is incompatible with verifiability, and we revise our definition in this paper.

election schemes from controlled-malleable encryption schemes satisfies ballot secrecy.

Related work. The concept of independence was introduced by Chor *et al.* [CGMA85] and studied in the context of election schemes by Gennaro [Gen95]. Cortier & Smyth [CS11,SC11,CS13] have discovered attacks on ballot secrecy in several election schemes and considered the relationship to independence [CS13, 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. Finally, proving that ballot secrecy can be satisfied by election schemes constructed from non-malleable encryption schemes has been shown by Bernhard, Pereira & Warinschi [BPW12b] and, in this paper, we generalise their result by proving that controlled-malleable encryption is sufficient.

2 Preliminaries

We adopt standard notation for the application of probabilistic algorithms A, namely, $A(x_1, \ldots, x_n; r)$ is the result of running A on input $x_1, \ldots, x_n$ and coins r. Moreover, $A(x_1, \ldots, x_n)$ denotes $A(x_1, \ldots, x_n; r)$, where r is chosen at random. We write $x \leftarrow \alpha$ for the assignment of α to x. In addition, we write $x \leftarrow_R S$ for the assignment of a random element from the set S 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 [DDN91, BDPR98, DDN00] 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. This notion can be captured by a pair of cryptographic games – namely, $\mathsf{Succ}^{\mathrm{CPA}}_{\mathcal{A},\Pi}$ and $\mathsf{Succ}^{\mathrm{CPA}}_{\mathcal{A},\Pi,\$}$ – between an adversary and a challenger. The first three steps of both games are identical. First, the challenger constructs a key pair (pk, sk). Secondly, the adversary $\mathcal{A}$ executes the algorithm A_1 on the public key pk and outputs the pair (M, s), where M is a sampling algorithm for some message space and s is some state information. Thirdly, the challenger randomly selects a plaintext x from the message space; at this point, the challenger in $\mathsf{Succ}^{\mathrm{CPA}}_{\mathcal{A},\Pi,\$}$ performs an additional step, namely, the challenger samples a second plaintext x'. Fourthly, the challenger constructs a ciphertext $y \leftarrow \mathsf{Enc}_{pk}(x)$. Fifthly, the adversary executes algorithm A_2 which outputs a relation R and a vector of ciphertexts $\mathbf{y}$. Finally, the challenger decrypts $\mathbf{y}$ and outputs the corresponding plaintexts $\mathbf{x}$. The encryption scheme satisfies non-malleability if the adversary's relation R cannot meaningfully relate x and $\mathbf{x}$. Formally, Definition 2 recalls the non-malleability game proposed by Bellare *et al.* [BDPR98].

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*

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

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

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

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

In the above games we insist that the message space is valid (that is, $|x| = |x'|$ *for any* $x, x' \leftarrow_R M$ *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* $\textsf{\textit{NM-CPA}}_{\mathcal{A},\Pi}(n) \leq \mathsf{negl}(n)$.

3 Election schemes and ballot secrecy

Based upon Bernhard *et al.* [BCP+11, BPW12b, BPW12a], we define a syntax for *election schemes* as follows.

Definition 3 (Election scheme)**.** *An* election scheme *is a tuple of efficient algorithms* $(\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ *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.*, the tally algorithm outputs auxiliary data aux, in addition to the election outcome, which can be used to store signatures of knowledge proving that the election result has been correctly computed from the bulletin board, for example. Moreover, our 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, we explicitly define the bulletin board and election result as multisets, make some minor changes to error handling, and merge some functionality into a single function[4].

We demonstrate the applicability of our definition by recalling the construction (Definition 4) for election schemes proposed by Bernhard *et al.* [BCP+11,

[4]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.

BPW12b]. We stress that more sophisticated schemes can also be captured – for example, Bernhard *et al.* [BCP$^+$11, BPW12b, BPW12a] model Helios – but the following scheme is sufficient for our purposes.

Definition 4 (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 NM-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[5].

Ballot Secrecy

Ballot secrecy is a *de facto* standard property of election schemes and, based upon Bernhard *et al.* [BCP$^+$11, BPW12b, BPW12a], we formalise a cryptographic game for ballot secrecy (Definition 5). We will describe the differences between 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 $\mathfrak{v}$ and auxiliary data aux as follows: if the honestly cast votes on the bulletin board $\mathfrak{bb}_0$ correspond to the honestly cast votes on the

[5] Definition 4 rectifies a mistake in the presentation by Bernhard, Pereira & Warinschi [BPW12b] 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.* [BCP$^+$11] avoid this problem in a similar fashion.)

bulletin board $\mathfrak{bb}_1$, then the challenger reveals the election result and associated auxiliary data for $\mathfrak{bb}_\beta$, otherwise, the challenger reveals the election result for $\mathfrak{bb}_0$ and auxiliary data $\perp$, 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 $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_\beta)$, if $L_0 = L_1$, and $aux \leftarrow \perp; (\mathfrak{v}, aux') \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_0)$, otherwise. Finally, the adversary executes the algorithm A_2 on the election result $\mathfrak{v}$, auxiliary data aux, 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 $\textit{IND-SEC}_{\mathcal{A},\Gamma}(n)$ be the quantity defined below, where n is the security parameter.*

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

In the above game, L_0 and L_1 are multisets, the oracle $\mathcal{O}$ is defined below, and $\mathfrak{v}$ and aux are defined as follows: if $L_0 = L_1$, then $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_\beta)$, otherwise, $aux \leftarrow \perp; (\mathfrak{v}, aux') \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_0)$.

- *$\mathcal{O}(v_0, v_1)$ 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 $\textit{IND-SEC}_{\mathcal{A},\Gamma}(n) \leq \mathsf{negl}(n)$.*

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. For example, suppose an adversary in control of one dishonest voter can violate ballot secrecy in a referendum with two honest voters, when: all voters participate, each voter casts a valid vote, and no auxiliary data is produced (as per the Enc2Vote construction, we can model the absence of auxiliary data using a constant symbol such as $\perp$). In this setting, we require a vote space $\{v_0, v_1\}$ and the adversary must make three oracle calls, namely, $\mathcal{O}(v_0, v_1)$, $\mathcal{O}(v_1, v_0)$, and $\mathcal{O}(b)$. It follows that the election result will be $\{v_0, v_1, v\}$, where v is the adversary's vote. Moreover, the adversary must have a strategy to generate b such that the adversary's vote v is related to either v_0 or v_1, otherwise, the election results from both bulletin boards will be equal and the adversary cannot win the game. We stress that a unanimous election result – for instance, the election result generated by tallying the bulletin board $\mathfrak{bb}_\beta$ produced by the oracle calls $\mathcal{O}(v_0, v_1)$, $\mathcal{O}(v_0, v_1)$, and $\mathcal{O}(b)$, where b contains the vote v_β – 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 our original presentation. Our original presentation of ballot secrecy [SB13, Definition 5] always outputs auxiliary data derived from tallying: we compute $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_\beta)$, if $L_0 = L_1$, and $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_0)$, otherwise. Galindo & Cortier [GC13] have shown that this definition is too strong, since it is incompatible with verifiability, in particular, verification will succeed if $\beta = 0$ and fail if $\beta = 1$, in the case $L_0 \neq L_1$. We overcome this limitation by weakening our original definition, in particular, we compute $aux \leftarrow \perp; (\mathfrak{v}, aux') \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_0)$, when $L_0 \neq L_1$.

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 [CS13, 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 [DDN91, DDN00]. Bellare *et al.* [BDPR98] build upon these results to introduce NM-CPA (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*

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

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

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

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

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 $v \leftarrow_R M$ 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 $\textsf{NM-BB}_{\mathcal{A},\Gamma}(n) \leq \mathsf{negl}(n)$.*

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.

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

In Proposition 1, 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 [BPW12b, Theorem 4.2].

More generally, we could adapt the non-malleability game for encryption (Definition 2) to a non-malleability game for ballots. In this setting, given an election scheme satisfying our non-malleability game for ballots and such that the bulletin board algorithm rejects duplicates, we believe that the election scheme satisfies ballot independence. Formalising this result is a possible direction for future research.

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 [BS99] 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 indistinguishability 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 7). 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 7 (IND-BB: Ballot independence)**.** *Let $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ be an election scheme, $\mathcal{A} = (A_1, A_2, A_3)$ be an adversary, n be the security parameter and $\textsf{IND-BB}_{\mathcal{A},\Gamma}(n)$ the cryptographic game defined below.*

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

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 $\textsf{IND-BB}_{\mathcal{A},\Gamma}(n) \leq \mathsf{negl}(n)$.*

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 [BS99] 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 1 ($\mathsf{NM\text{-}BB} = \mathsf{IND\text{-}BB}$)**.** *Given an election scheme Γ, we have Γ satisfies $\mathsf{NM\text{-}BB}$ if and only if Γ satisfies $\mathsf{IND\text{-}BB}$.*

Theorem 1 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.

Proof. Let $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$. For the forward implication, suppose Γ does not satisfy $\mathsf{IND\text{-}BB}$, hence, for any negligible function f, there exists an adversary $\mathcal{A} = (A_1, A_2, A_3)$ and a security parameter n such that $\mathsf{IND\text{-}BB}_{\mathcal{A},\Gamma}(n) > f(n)$, moreover, $\mathsf{IND\text{-}BB}_{\mathcal{A},\Gamma}(n) > 2 \cdot f(n)$, since doubling a negligible function produces another negligible function. Let us show that Γ does not satisfy $\mathsf{NM\text{-}BB}$, by constructing an adversary $\mathcal{B} = (B_1, B_2)$ as follows:

Algorithm B_1**.** Given input $\mathfrak{m}$ and pk, the algorithm computes $(v_0, v_1, s) \leftarrow A_1^{\mathcal{O}}(\mathfrak{m}, pk)$ and outputs $(\{v_0, v_1\}, (\{v_0, v_1\}, s))$.

Algorithm B_2**.** Given input $(\{v_0, v_1\}, s)$, the algorithm computes $t \leftarrow A_2^{\mathcal{O}}(s)$, selects some random coins r, and outputs the relation R such that $R(v, \mathfrak{v})$ holds if $v = v_g$ and fails otherwise, where $g \leftarrow A_3(t, \mathfrak{v}; r)$.

Let us consider executions of $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{B},\Pi}(n)$ and $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{B},\Pi,\$}(n)$.

- First, $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{B},\Pi}(n)$, where a single vote v is sampled from M. By inspecting the values provided to the embedded instance of $\mathcal{A}$, we see that the distribution of these values is identical to if $\mathcal{A}$ were interacting with $\mathsf{IND\text{-}BB}$ directly. The use of A_3 is in a non-black-box manner but this does not matter: it is still invoked exactly one time in the game. Hence, the probability that A_3's output matches the challenger's bit β is equal to the probability that $\mathcal{A}$ wins the $\mathsf{IND\text{-}BB}$ game, that is, strictly greater than $(2 \cdot f(n) + 1)/2$.
- Secondly, $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{B},\Pi,\$}(n)$, where two votes v and v' are sampled from M. The value v is independent of A's perspective, indeed, v could be sampled after A_3 has terminated and immediately before evaluating the relation R. It follows immediately that R holds iff $v = v_g$, where g is A_3's output and g is independent of v. Hence, the probability that R holds is $1/2$.

The advantage of our adversary $\mathcal{B}$ in $\mathsf{NM\text{-}BB}$ is therefore strictly greater than $(2 \cdot f(n)+1)/2-1/2 = f(n)$, concluding this direction of the proof by contraposition.

For the reverse implication, suppose Γ does not satisfy $\mathsf{NM\text{-}BB}$, hence, for any negligible function f there exists an adversary $\mathcal{A} = (A_1, A_2)$ and a security parameter n such that $\mathsf{NM\text{-}BB}_{\mathcal{A},\Gamma}(n) > 2 \cdot f(n)$. Let us construct an adversary $\mathcal{B} = (B_1, B_2, B_3)$ against $\mathsf{IND\text{-}BB}$ as follows:

Algorithm B_1**.** Given input $\mathfrak{m}$ and pk, the algorithm computes $(M, s) \leftarrow A_1^{\mathcal{O}}(\mathfrak{m}, pk)$; $v_0, v_1 \leftarrow M$ and outputs $(v_0, v_1, (v_0, M, s))$.

Algorithm B_2**.** Given input (v_0, M, s), the algorithm computes $R \leftarrow A_2^{\mathcal{O}}(M, s)$ and outputs (v_0, R).

Algorithm B_3**.** Given input (v_0, R) and $\mathfrak{v}$, the algorithm evaluates $R(v_0, \mathfrak{v})$ and if the relation holds, then the algorithm outputs 0, otherwise, the algorithm outputs 1.

If the challenger selects $\beta = 0$ in $\mathsf{IND\text{-}BB}$, then the embedded adversary $\mathcal{A}$ sees exactly the same distribution of values as in $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{B},\Pi}(n)$, otherwise ($\beta = 1$), $\mathcal{A}$ sees the same distribution as in the second $\mathsf{Succ}^{\mathrm{BB}}_{\mathcal{B},\Pi,\$}(n)$. Let g be $\mathcal{B}$'s guess in $\mathsf{IND\text{-}BB}$. The success probability of B is:

$$\begin{aligned}
\Pr[\beta = g] &= \Pr[\beta = 0] \cdot \Pr[g = 0 \mid \beta = 0] + \Pr[\beta = 1] \cdot \Pr[g = 1 \mid \beta = 1] \\
&= 1/2 \cdot (\Pr[g = 0 \mid \beta = 0] + \Pr[g = 1 \mid \beta = 1]) \\
&= 1/2 \cdot (\Pr[R(v_0, \mathfrak{v})] + (1 - \Pr[R(v_1, \mathfrak{v})])) \\
&= 1/2 + 1/2 \cdot \mathsf{NM\text{-}CPA}_{\mathcal{A},\Pi}(n)
\end{aligned}$$

Since $1/2 + 1/2 \cdot \mathsf{NM\text{-}CPA}_{\mathcal{A},\Pi}(n) > 1/2 + f(n)$, the advantage of B is greater than $f(n)$, concluding the proof. $\square$

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 1, 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*[6], denoted $\mathsf{NM\text{-}CPA}/R$ (pronounced "NM-CPA modulo R"), which we will show is sufficient for ballot independence.

Definition 8 (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*

[6]The term is taken from Chase et al. [CKLM12] who introduce controlled malleability for zero-knowledge proofs.

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

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* $\mathsf{Dec}_{sk}(\mathbf{c})$*.*

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

Definition 9 (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 10), 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 2).

Definition 10 (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 2. *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 2 is similar to the proof of [BPW12b, Theorem 4.2].

Intuitively, we could adapt the controlled malleability game for encryption (Definition 8) to a controlled malleability game for ballots. In this setting, given an election scheme satisfying our controlled malleability game for ballots and such that the bulletin board algorithm rejects duplicates, we believe that the election scheme satisfies ballot independence. Moreover, the generalised definition would allow us to consider whether controlled malleability for ballots is necessary for ballot independence. (Clearly such results cannot be considered using controlled malleability for encryption, since this definition excludes election schemes based upon alternative cryptographic primitives, such as commitments, for example.) Formalising this result is a possible direction for future research.

4.4 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 (Section 4.3). 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.* [BCP+11,BPW12b,BPW12a] 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 [SC10, §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 2 (Ballot secrecy implies ballot independence)**.** *Given an election scheme* $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ *satisfying ballot secrecy, we have* Γ *satisfies ballot independence.*

Theorem 2 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 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 2 is standard: by contradiction, we construct an adversary $\mathcal{B} = (B_1, B_2)$ against $\mathsf{IND\text{-}SEC}$ from a successful adversary $\mathcal{A} = (A_1, A_2, A_3)$ against $\mathsf{IND\text{-}BB}$ such that $\mathcal{B}$ ensures $\mathcal{A}$'s perspective of the bulletin board and election result are consistent with $\mathsf{IND\text{-}BB}$. Before explaining how we ensure that $\mathcal{A}$'s perspective is consistent, let us briefly review the distinction between $\mathcal{A}$'s and $\mathcal{B}$'s perspectives of their respective bulletin board and election result.

- In $\mathsf{IND\text{-}BB}$, the adversary A_1 expects $\mathcal{O}_{\mathcal{A}}() = \mathfrak{bb}$ such that $b \in \mathfrak{bb}$ implies A_1 previously called $\mathcal{O}_{\mathcal{A}}(b)$. Moreover, adversaries A_2 and A_3 expect $\mathcal{O}_{\mathcal{A}}() = \mathfrak{bb} \cup \{b'\}$ such that b' is the challenge ballot and $b \in \mathfrak{bb}$ implies A_1 or A_2 previously called $\mathcal{O}_{\mathcal{A}}(b)$. Furthermore, A_3 observes the election result $\mathfrak{v}\backslash\{v_\beta\}$, where v_β is the challenge vote and $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb})$.

- By comparison, in $\mathsf{IND\text{-}SEC}$, the adversary $\mathcal{B}$ expects $\mathcal{O}_{\mathcal{B}}() = \mathfrak{bb}$ such that $b \in \mathfrak{bb}$ implies $\mathcal{B}$ previously called $\mathcal{O}_{\mathcal{B}}(b)$ or $\mathcal{O}_{\mathcal{B}}(v_0, v_1)$, and in the latter case the oracle constructed $b = \mathsf{Vote}_{pk}(v_\beta)$. Furthermore, $\mathcal{B}$ observes the election result $\mathfrak{v}$, where $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_\beta)$, if $L_0 = L_1$, and $(\mathfrak{v}, aux') \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_0)$, otherwise.

It follows immediately that A_2 and A_3 will observe a challenge ballot on the bulletin board, whereas, $\mathcal{B}$ will not. In addition, the challenge vote will be removed from the election result observed by A_3, whereas no votes are removed from the election result observed by $\mathcal{B}$. Let us now informally explain how $\mathcal{B}$ ensures that $\mathcal{A}$'s perspective of the bulletin board and election result are consistent with $\mathsf{IND\text{-}BB}$. First, $\mathcal{B}$ ensures that a challenge ballot appears on the bulletin board observed by adversaries A_2 and A_3 by calling $\mathcal{O}_{\mathcal{B}}(v_0, v_1)$, where votes v_0 and v_1 are output by A_1. Secondly, the adversary $\mathcal{B}$ calls $\mathcal{O}_{\mathcal{B}}(v_1, v_0)$ and inputs the election result $\mathfrak{v}\backslash\{v_1, v_0\}$ to A_3, where $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb})$. We remark that tallying after the first step will produce an election result which includes the challenge vote v_β and does not correspond to the election result expected by $\mathcal{A}$; the second step overcomes this problem.

Proof of Theorem 2. Suppose $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ is an election scheme with ballot secrecy that does not satisfy $\mathsf{IND\text{-}BB}$, hence for any negligible function f there exists an adversary $\mathcal{A} = (A_1, A_2, A_3)$ and security parameter n such that $\mathsf{IND\text{-}BB}_{\mathcal{A},\Gamma}(n) > f(n)$. We construct an adversary $\mathcal{B} = (B_1, B_2)$ against $\mathsf{IND\text{-}SEC}$ as follows.

Algorithm B_1**.** Given input $\mathfrak{m}$ and pk, the algorithm proceeds as follows. First, B_1 computes $(v_0, v_1, s) \leftarrow A_1^{\mathcal{O}_{\mathcal{A}}}(\mathfrak{m}, pk)$, handling any oracle calls from A_1 as follows: if A_1 calls $\mathcal{O}_{\mathcal{A}}(b)$, then B_1 calls $\mathcal{O}_{\mathcal{B}}(b)$, similarly, if A_1 calls $\mathcal{O}_{\mathcal{A}}()$, then B_1 computes $\mathfrak{bb} \leftarrow \mathcal{O}_{\mathcal{B}}()$ and returns $\mathfrak{bb}$ to A_1. Secondly, B_1 creates the challenge ballot and adds it to the bulletin board by computing $\mathcal{O}_{\mathcal{B}}(v_0, v_1)$. Thirdly, B_1 computes $t \leftarrow A_2^{\mathcal{O}_{\mathcal{A}}}(s)$, handling any oracle calls from A_2 as before. Finally, B_1 computes $\mathcal{O}_{\mathcal{B}}(v_1, v_0)$; and outputs t.

Algorithm B_2**.** Given input $\mathfrak{v}$, aux and t, the algorithm computes $A_3(t, \mathfrak{v} \setminus \{v_0, v_1\})$ and outputs A_3's guess.

The embedded adversary $\mathcal{A}$ sees the same distribution of all elements as in the $\mathsf{IND\text{-}BB}$ game for the same value of β. Indeed, the challenge ballot is computed in the same manner, $\mathcal{O}_{\mathcal{A}}()$ produces the expected multiset of ballots (we stress that the ballot introduced in the final step of B_1 — to ensure that the election result is consistent with A_3's expectations – never appears in a multiset output by $\mathcal{O}_{\mathcal{A}}()$, since this ballot is added to the bulletin board after all oracle calls by $\mathcal{A}$), and the election result observed by A_3 is as expected. It follows that $\mathcal{B}$ guesses β correctly with the same advantage as $\mathcal{A}$ and, therefore, $\mathsf{IND\text{-}SEC}_{B,\Gamma}(n) > f(n)$, concluding our proof. □

5.1 Critique of Desmedt & Chaidos's Helios variant

Intuitively, Theorem 2 contradicts the results by Desmedt & Chaidos [DC12], 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 [CS11, SC11, CS13] – 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 [DC12, 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 [Cha13].

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 [CS13, 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 2, where we take the election result and challenge vote, and compute the partial election result which removes the challenge vote. 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 [BY86, HK02, HK04, DK05]. 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 shows that ballot secrecy is stronger than independence and Footnote 5 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 denying the adversary access to auxiliary data we can show that the two games are equivalent (Theorem 3) 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 3 ($\mathsf{NM\text{-}BB} = \mathsf{IND\text{-}SEC}$, 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* $(\mathfrak{bb}_0, \mathfrak{m}, pk, sk) \leftarrow \mathsf{Setup}(1^n)$*, multisets* $\mathfrak{bb}$ *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.*

Proof. Suppose Γ is an election scheme that does not satisfy IND-BB, hence for any negligible function f there exists an adversary $\mathcal{A}$ and security parameter n such that $\textsf{IND-BB}_{\mathcal{A},\Gamma}(n) > f(n)$. The adversary $\mathcal{B}$ defined in the proof of Theorem 2 is such that $\textsf{IND-SEC}_{B,\Gamma}(n) > f(n)$. It remains to show that ballot independence implies ballot secrecy.

Suppose Γ is an election scheme that does not satisfy IND-SEC, hence for any negligible function f there exists an adversary $\mathcal{B} = (B_1, B_2)$ and security parameter n such that $\textsf{IND-SEC}_{\mathcal{B},\Gamma}(n) > f(n)$. The probability of $\textsf{IND-SEC}_{\mathcal{B},\Gamma}(n) > f(n)$ without making any two-element oracle calls, with input from the voting scheme's vote space, is $1/2$, since IND-SEC with $\beta = 0$ is identical to IND-SEC with $\beta = 1$ in this case. Accordingly, without loss of generality, we can assume that B_1 makes at least one such two-element oracle call (in cases where the assumption does not hold, we let $\mathcal{A}$ guess β randomly, without losing any of $\mathcal{B}$'s advantage). We shall use $\mathcal{B}$ to construct an adversary $\mathcal{A} = (A_1, A_2, A_3)$ Let q be an upper bound on the number of two-element oracle calls made by B_1. We can assume that q is polynomial in the security parameter, because $\mathcal{B}$ is efficient. We proceed by introducing hybrid games $G_0, \ldots, G_q$. For $0 \leq i \leq q$ let G_i be the game defined below.

$$(\mathfrak{bb}_0, \mathfrak{m}, pk, sk) \leftarrow \mathsf{Setup}(1^n);\ \mathfrak{bb}_1 \leftarrow \mathfrak{bb}_0;\ s \leftarrow B_1^{\mathcal{O}}(\mathfrak{m}, pk);\ g \leftarrow B_2(\mathfrak{v}, \perp, s);\ \text{output } g$$

In the above game, the oracle $\mathcal{O}$ is defined below, and $\mathfrak{v}$ is defined as follows, namely, $(\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_0)$.

- $\mathcal{O}(v_0, v_1)$ executes $b_0 \leftarrow \mathsf{Vote}_{pk}(v_0); b_1 \leftarrow \mathsf{Vote}_{pk}(v_k); \mathfrak{bb}_0 \leftarrow \mathsf{BB}(\mathfrak{bb}_0, b_0)$; $\mathfrak{bb}_1 \leftarrow \mathsf{BB}(\mathfrak{bb}_1, b_1)$, where $k = 1$ for the first i queries and $k = 0$ for any subsequent query.
- $\mathcal{O}(b)$ assigns $\mathfrak{bb}_1' \leftarrow \mathfrak{bb}_1$, executes $\mathfrak{bb}_1 \leftarrow \mathsf{BB}(\mathfrak{bb}_1, b)$ and if $\mathfrak{bb}_1 \neq \mathfrak{bb}_1'$, then executes $\mathfrak{bb}_0 \leftarrow \mathsf{BB}(\mathfrak{bb}_0, b)$.
- $\mathcal{O}()$ outputs $\mathfrak{bb}_1$.

We insist that two-element oracle queries always provide inputs from $\mathfrak{m}$.

We demonstrate that the adversary $\mathcal{B}$'s perspective in G_0 is equivalent to $\mathcal{B}$'s perspective in the IND-SEC game when $\beta = 0$. The inputs to B_1 can trivially be observed to be equivalent in both instances, because they are generated by Setup. Moreover, B_1's oracle access is equivalent in each case, because $\mathfrak{bb}_0$ in IND-SEC is equivalent to $\mathfrak{bb}_1$ in G_0. It follows that B_1's output is equivalent in both settings. Furthermore, the tallies generated in both G_0 and IND-SEC are equivalent, because $\mathfrak{bb}_0$ and $\mathfrak{bb}_1$ are equivalent in G_0. Since our hypothesis asserts that the auxilliary data output by tallying is a constant symbol, it follows that the inputs to B_2 are equivalent in both instances.

Similarly, we demonstrate that the adversary $\mathcal{B}$'s perspective in G_q is equivalent to $\mathcal{B}$'s perspective in the IND-SEC game when $\beta = 1$. As before, the inputs to B_1 can trivially be observed to be equivalent in both instances. Moreover, since q is an upper bound on the number of two-element oracle calls made by B_1, the oracle definitions in G_q and $\mathsf{IND\text{-}SEC}$ are identical, with the exception of updating L_0 and L_1 in $\mathsf{IND\text{-}SEC}$ (which does not influence B_1's perspective). Once again, it follows that B_1's output is equivalent in both settings. Furthermore, the tallies generated in both G_0 and $\mathsf{IND\text{-}SEC}$ are equivalent, in particular, G_0 tallies $\mathfrak{bb}_0$, which is equivalent to either of the following cases: 1) tallying $\mathfrak{bb}_1$ in $\mathsf{IND\text{-}SEC}$ when $L_0 = L_1$, because $\mathfrak{bb}_1$ is equivalent to $\mathfrak{bb}_0$ in this case; or 2) tallying $\mathfrak{bb}_0$ in $\mathsf{IND\text{-}SEC}$. As before, we conclude that the inputs to B_2 are equivalent in both instances.

It follows that $\mathcal{B}$'s advantage against $\mathsf{IND\text{-}SEC}$ is $\mathcal{B}$'s distinguishing advantage between G_0 and G_q. Moreover, since $\mathcal{B}$ has non-negligible advantage of distinguishes G_0 and G_q, there exists an integer i such that $0 \leq i < q$ and $\mathcal{B}$ distinguishes G_i and G_{i+1} with non-negligible advantage, more precisely, we have the following fact.

Fact 1. *The adversary $\mathcal{B}$ distinguishes G_i and G_{i+1} with probability greater than $\mathsf{IND\text{-}SEC}_{\mathcal{B},\Gamma}(n)/q$ for some integer i such that $0 \leq i < q$.*

Proof of Fact 1. Let p_i be the probability that the adversary $\mathcal{B}$ outputs 1 following an interaction with G_i, where $0 \leq i \leq q$. The probability is taken over all random choices in this experiment. Since $\mathcal{B}$ is an $\mathsf{IND\text{-}SEC}$ adversary with a non-negligible distinguishing probability, the quantity $\mathsf{IND\text{-}SEC}_{\mathcal{B},\Gamma}(n) = |p_q - p_0|$ is non-negligible. We write this as a telescope sum:

$$|p_q - p_0| = |(p_q - p_{q-1}) + (p_{q-1} - p_{q-2}) + \ldots + (p_1 - p_0)|$$

Repeatedly applying the inequality $|a + b| \leq |a| + |b|$ we find:

$$\mathsf{IND\text{-}SEC}_{\mathcal{B},\Gamma}(n) \leq |(p_q - p_{q-1})| + |(p_{q-1} - p_{q-2})| + \ldots + |(p_1 - p_0)|$$

The largest of the quantities on the right-hand side must therefore be at least $\mathsf{IND\text{-}SEC}_{\mathcal{B},\Gamma}(n)/q$, concluding the proof of Fact 1.

Using Fact 1 we proceed the proof of Theorem 3. By Fact 1, let i be an integer such that $\mathcal{B}$ distinguishes G_i and G_{i+1} with probability greater than $\mathsf{IND\text{-}SEC}_{\mathcal{B},\Gamma}(n)/q$, where $0 \leq i < q$. The probability of $\mathcal{B}$ distinguishing games G_i and G_{i+1} with fewer than than $i+1$ two-element oracle queries is $1/2$, since the two games are identical until the $i+1$ such query. Accordingly, without loss of generality, we can assume that $\mathcal{B}$ makes at least $i+1$ two-element oracle queries and construct the adversary $\mathcal{A}$ as follows (in cases where there are fewer than $i+1$ queries, we can let $\mathcal{A}$ guess randomly).

Algorithm A_1**.** Given input $\mathfrak{m}$ and pk, A_1 initialises multisets $L_0 \leftarrow \emptyset$ and $L_1 \leftarrow \emptyset$, and runs $B_1^{\mathcal{O}_\mathcal{B}}(\mathfrak{m}, pk)$. Oracle calls by B_1 are handled as follows.

- $\mathcal{O}_\mathcal{B}(b)$: A_1 calls $\mathcal{O}_\mathcal{A}(b)$.
- $\mathcal{O}_\mathcal{B}()$: A_1 computes $\mathfrak{bb} \leftarrow \mathcal{O}_\mathcal{A}()$ and returns $\mathfrak{bb}$ to B_1.

- $\mathcal{O}_{\mathcal{B}}(v_0, v_1)$: For the first i calls, A_1 computes

$$b \leftarrow \mathsf{Vote}_{pk}(v_1); \mathcal{O}_{\mathcal{A}}(b); L_0 \leftarrow L_0 \cup \{v_0\}; L_1 \leftarrow L_1 \cup \{v_1\}$$

For call $i+1$, A_1 suspends B_1 and saves its state as t, and outputs $(v_0, v_1, (t, v_0, L_0, L_1))$ to the challenger.

(A_1 terminates on the i+1-st two-element oracle query.)

Algorithm A_2**.** Given input (t, v_0^c, L_0, L_1), A_2 resumes B_1 with state t. Oracle calls by B_1 are handled as above, except calls $\mathcal{O}_{\mathcal{B}}(v_0, v_1)$, which are handled as follows: $b \leftarrow \mathsf{Vote}_{pk}(v_0); \mathcal{O}_{\mathcal{A}}(b)$. When B_1 outputs some state s, A_2 returns (s, v_0^c, L_0, L_1).

Algorithm A_3**.** Given input (s, v_0^c, L_0, L_1) and $\mathfrak{v}$, A_3 assigns $\mathfrak{v}' \leftarrow \{v_0^c\} \cup L_0 \cup (\mathfrak{v} \setminus L_1)$, computes $g \leftarrow B_2(\mathfrak{v}', \perp, s)$, and outputs g. (Informally, the assignment computes the result $\mathfrak{v}'$ by replacing all the votes that came from two element oracle calls made by B_1 – namely, the votes in the multiset L_1 – with the votes in the multiset L_0.)

This construction provides a view of either G_i or G_{i+1} towards $\mathcal{B}$, in particular, $\mathcal{O}_{\mathcal{B}}(v_0, v_1)$ queries compute $\mathfrak{bb}_\beta \leftarrow \mathsf{BB}(\mathfrak{bb}_\beta, b)$, where β is choosen by the challenger and b is defined as follows: $b \leftarrow \mathsf{Vote}_{pk}(v_1)$ for the first i queries, $b \leftarrow \mathsf{Vote}_{pk}(v_\beta)$ for the $i+1$ query, and $b \leftarrow \mathsf{Vote}_{pk}(v_0)$ for any subsequent queries. Moreover, $\mathfrak{v}'$ is computed as if $\beta = 0$. It follows that the construction provides a view of G_i, if $\beta = 0$, and G_{i+1}, otherwise (i.e., $\beta = 1$). We preserve the distinguishing advantage $f(n)$ of $\mathcal{B}$ in our adversary $\mathcal{A}$ against IND-BB. □

The ESORICS'13 version of this paper suggests circumstances under which Theorem 3 could be generalised: we hinted that a stronger notion of ballot secrecy coincides with ballot independence for zero-knowledge auxiliary data [SB13, Remark 16]. Unfortunately, such a result cannot hold, because we have seen that the stronger notion of ballot secrecy is incompatible with verifiability (Section 3), whereas ballot independence is compatible with verifiability, i.e., verifiable election schemes with zero-knowledge auxiliary data satisfy ballot independence but not the strong notion of ballot secrecy. Considering whether NM-BB = IND-SEC for zero-knowledge auxiliary data is a possible direction for future work.

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. Furthermore, we provide some sufficient conditions for ballot independence and, hence, ballot secrecy: we show that non-malleable ballots are sufficient for independence and secrecy, and introduce a weaker notion of controlled-malleable encryption which is also 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 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).

References

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

[AMPQ09] Ben Adida, Olivier de Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios. In *EVT/WOTE'09: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. USENIX Association, 2009.

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

[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.

[BGP11] Philippe Bulens, Damien Giry, and Olivier Pereira. Running Mixnet-Based Elections with Helios. In *EVT/WOTE'11: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. USENIX Association, 2011.

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

[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.

[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.

[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.

[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.

[Cha13] Pyrros Chaidos. Private email communication, March/April 2013.

[CKLM12] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable Proof Systems and Applications. In *Advances in Cryptology — Eurocrypt 2012*, volume 7237 of *LNCS*, pages 281–300, 2012.

[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.

[DK05] Yvo Desmedt and Kaoru Kurosawa. Electronic Voting: Starting Over? In *ISC5: International Conference on Information Security*, volume 3650 of *LNCS*, pages 329–343. Springer, 2005.

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

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

[GC13] David Galindo and Véronique Cortier. Private email communication, August 2013.

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

[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.

[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.

[SC10] Ben Smyth and Véronique Cortier. Does Helios ensure ballot secrecy? Cryptology ePrint Archive, Report 2010/625 (version 20101217:132825), 2010.

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

[Sch13] Bruce Schneier. Hacking the Papal Election. `https://www.schneier.com/blog/archives/2013/02/hacking_the_pap.html`, 2013.