Ballot secrecy with malicious bulletin boards

Ben Smyth[1] and David Bernhard[2]

[1]Mathematical and Algorithmic Sciences Lab, France Research Center, Huawei Technologies Co. Ltd., France
[2]University of Bristol, UK

December 14, 2014

Abstract

This letter proposes a formal definition of ballot secrecy in the computational model of cryptography. The definition builds upon and strengthens earlier definitions by Bernhard *et al.* (ASIACRYPT'12, ESORICS'11 & ESORICS'13). The new definition is intended to ensure that ballot secrecy is preserved in the presence of malicious bulletin boards, whereas earlier definitions by Bernhard *et al.* only consider honest bulletin boards.

1 Introduction

Voters should be able to express their free-will in elections without fear of retribution; this property is known as privacy. *Ballot secrecy*[1] has emerged as a *de facto* standard privacy requirement of election schemes.

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

Bernhard *et al.* [SB14,SB13,BPW12a,BPW12b,BCP$^+$11] formally define ballot secrecy in the computational model of cryptography. Their definitions assume the bulletin board is honest and provide no privacy guarantees if this trust assumption is violated. This letter builds upon and strengthens the definitions by Bernhard *et al.* to ensure that ballot secrecy is preserved in the presence of malicious bulletin boards.

2 Preliminaries

Standard notation is adopted 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

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

coins r. Moreover, $A(x_1, \ldots, x_n)$ denotes $A(x_1, \ldots, x_n; r)$, where r is chosen at random. The assignment of α to x is written $x \leftarrow \alpha$ and the assignment of a random element from set S to x is written $x \leftarrow_R S$. Vectors are denoted using boldface, for example, $\mathbf{x}$. Set membership notation is extended to vectors: x is an element (respectively, x is not an element) of the vector $\mathbf{x}$ is written $x \in \mathbf{x}$ (respectively, $x \notin \mathbf{x}$).

The syntax and security definitions for election schemes are recalled[2] from Smyth & Bernhard [SB14, SB13]:

Definition 1 (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 set 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 set. 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 set. 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* $v \in \mathfrak{m}$*, sets* $\mathfrak{bb}$*, 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* $(\mathfrak{v}', aux') \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}')$*, it holds 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,* $\mathfrak{v}' = \emptyset$*.*

Definition 2 (Ballot secrecy with a trusted bulletin board)**.** *Let* $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ *be an election scheme,* $\mathcal{A} = (A_1, A_2)$ *be an adversary, and* $\textsf{\textit{IND-SEC}}_{\mathcal{A},\Gamma}(n)$ *be the quantity defined below, where* n *is the security parameter.*

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

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,* $aux \leftarrow \perp; (\mathfrak{v}, aux') \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}_0)$*.*

[2] The definitions assume that the bulletin board is a set – rather than a multiset, *à la* Smyth & Bernhard – to prevent the construction of election schemes which are vulnerable to ballot secrecy attacks, when the bulletin board is a multiset [CS11, CS13].

- $\mathcal{O}(v_0, v_1)$ *computes* $L_0 \leftarrow L_0 \cup \{v_0\}; L_1 \leftarrow L_1 \cup \{v_1\}; b_0 \leftarrow \mathsf{Vote}_{pk}(v_0); b_1 \leftarrow \mathsf{Vote}_{pk}(v_1); \mathfrak{bb}_0 \leftarrow \mathsf{BB}(\mathfrak{bb}_0, b_0); \mathfrak{bb}_1 \leftarrow \mathsf{BB}(\mathfrak{bb}_1, b_1)$*, where* $v_0, v_1 \in \mathfrak{m}$*.*

- $\mathcal{O}(b)$ *computes* $\mathfrak{bb}'_\beta \leftarrow \mathfrak{bb}_\beta; \mathfrak{bb}_\beta \leftarrow \mathsf{BB}(\mathfrak{bb}_\beta, b)$ *and if* $\mathfrak{bb}_\beta \neq \mathfrak{bb}'_\beta$*, then also computes* $\mathfrak{bb}_{1-\beta} \leftarrow \mathsf{BB}(\mathfrak{bb}_{1-\beta}, b)$*.*

- $\mathcal{O}()$ *outputs* $\mathfrak{bb}_\beta$*.*

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

The use of algorithm BB in Definition 2 implies that real-world elections must use this algorithm to ensure privacy. This may introduce an unnecessary trust assumption: voters must trust the system to only add ballots to the bulletin board using algorithm BB. The next section proposes a new definition of ballot secrecy that does not use this algorithm.

3 Ballot secrecy with malicious bulletin boards

A stronger definition of ballot secrecy is proposed:

Definition 3 (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.*

$$2 \cdot Pr[(\mathfrak{bb}, \mathfrak{m}, pk, sk) \leftarrow \mathsf{Setup}(1^n);\ \beta \leftarrow_R \{0,1\};\ L \leftarrow \emptyset;$$
$$(\mathfrak{bb}', s) \leftarrow A_1^{\mathcal{O}}(\mathfrak{bb}, \mathfrak{m}, pk);\ (\mathfrak{v}, aux) \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}') :$$
$$\{v_0 \mid b \in \mathfrak{bb}' \wedge (b, v_0, v_1) \in L\} = \{v_1 \mid b \in \mathfrak{bb}' \wedge (b, v_0, v_1) \in L\}$$
$$\wedge\ A_2(\mathfrak{v}, aux, s) = \beta] - 1$$

Oracle $\mathcal{O}$ *is defined as follows:*

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

Election scheme 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* $\textit{IND-SEC}^{\#}_{\mathcal{A},\Gamma}(n) \leq \mathsf{negl}(n)$*.*

Informally, the above game proceeds as follows. First, the challenger executes the setup algorithm to construct a bulletin board $\mathfrak{bb}$, a vote space $\mathfrak{m}$, a public key pk, and a private key sk. The challenger also selects a random bit β and initialises L as the empty set. Secondly, the adversary executes the algorithm A_1. The algorithm A_1 has access to an oracle $\mathcal{O}$ which outputs challenge ballots as follows: $\mathcal{O}(v_0, v_1)$ records chosen votes v_0 and v_1, and outputs a ballot for candidate v_β. Thirdly, the challenger computes the election result $\mathfrak{v}$ and auxiliary data aux. The challenger requires that the tallies of chosen votes are equivalent, thus preventing the adversary from trivially revealing β. (The distinction between $\beta = 0$ and $\beta = 1$ is trivial when the tallies of chosen votes differ, because the adversary can test for the presence of chosen votes in the election result.) Formally, equivalence between the tallies of chosen votes is captured by equality of the multisets $\{v_0 \mid b \in \mathfrak{bb}' \wedge (b, v_0, v_1) \in L\}$ and $\{v_1 \mid b \in \mathfrak{bb}' \wedge (b, v_0, v_1) \in L\}$. 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 challenge ballots she interacted with. Intuitively, if the adversary loses the game, then the adversary is unable to distinguish between ballots for different candidates, hence, voters' votes cannot be revealed. On the other hand, if the adversary wins the game, then there exists a strategy to distinguish ballots for different candidates.

Theorem 1. *If an election scheme satisfies ballot secrecy, then the election scheme satisfies ballot secrecy with a trusted bulletin board.*

The proof of Theorem 1 appears in Appendix A.

The inverse of Theorem 1 does not hold, as a variant of Bernhard *et al.*'s Backdoor-Enc2Vote construction [SB14,SB13,BPW12b,BCP+11] demonstrates:

Definition 4 (Backdoor-Enc2Vote)**.** *Given an asymmetric encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, the election scheme* $\mathsf{Backdoor\text{-}Enc2Vote}(\Pi)$ *is defined 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* $b \in \mathfrak{bb} \cup \{\perp\}$*, 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. If* $\perp \in \mathfrak{bb}$*, then* $aux \leftarrow \{(b, \mathsf{Dec}_{sk}(b)) \mid b \in \mathfrak{bb}\}$*, otherwise,* $aux \leftarrow \perp$*. It outputs the multiset* $\{\mathsf{Dec}_{sk}(b) \mid b \in \mathfrak{bb}\}$ *and auxiliary data* aux*.*

Intuitively, given an asymmetric encryption scheme Π satisfying NM-CPA, the construction $\mathsf{Backdoor\text{-}Enc2Vote}(\Pi)$ preserves ballot secrecy from Π until tallying. Moreover, if the bulletin board does not contain $\perp$, then algorithm Tally maintains ballot secrecy by returning the number of votes for each candidate as an unordered multiset of votes. However, if the bulletin board contains $\perp$, then the auxiliary data produced by algorithm Tally maps ballots to votes. Algorithm BB prevents $\perp$ from appearing on the bulletin board, hence, $\mathsf{Backdoor\text{-}Enc2Vote}(\Pi)$ preserves ballot secrecy with a trusted bulletin board. However, a malicious bulletin board may not use algorithm BB and, hence, ballot secrecy is not preserved:

Proposition 1. *Given an encryption scheme* Π *satisfying NM-CPA, the election scheme* $\mathsf{Backdoor\text{-}Enc2Vote}(\Pi)$ *satisfies ballot secrecy with a trusted bulletin board, but not ballot secrecy.*

A proof that $\mathsf{Backdoor\text{-}Enc2Vote}(\Pi)$ satisfies ballot secrecy with a trusted bulletin board can be constructed similarly to the proof of [BPW12b, Theorem 4.2]. And a proof that $\mathsf{Backdoor\text{-}Enc2Vote}(\Pi)$ does not satisfy ballot secrecy can be constructed by formalising an adversary that adds $\perp$ to the bulletin board.

4 Conclusion

This letter shows that malicious bulletin boards can violate privacy in a manner that cannot be detected by Bernhard *et al.*'s definitions of ballot secrecy. This problem is overcome by proposing a stronger definition of ballot secrecy.

Acknowledgements. We are particularly grateful to Susan Thomson for discussion that helped simplify our new definition of ballot secrecy. 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 ERC Advanced Grant ERC-2010-AdG-267188-CRIPTO. This work was performed in part at INRIA.

A Proof of Theorem 1

Suppose $\Gamma = (\mathsf{Setup}, \mathsf{Vote}, \mathsf{BB}, \mathsf{Tally})$ is an election scheme that does not satisfy ballot secrecy with a trusted bulletin board. By Definition 2, for all negligible functions negl, there exists a probabilistic polynomial-time adversary $\mathcal{A} = (A_1, A_2)$ and security parameter n such that $\mathsf{IND\text{-}SEC}_{\mathcal{A},\Gamma}(n) > \mathsf{negl}(n)$. An adversary $\mathcal{B} = (B_1, B_2)$ against $\mathsf{IND\text{-}SEC}^{\#}$ is constructed below. Let $\mathcal{O}_\mathcal{A}$ denote $\mathcal{A}$'s oracle and $\mathcal{O}_\mathcal{B}$ denote $\mathcal{B}$'s oracle.

Algorithm B_1. On input $\mathfrak{bb}$, $\mathfrak{m}$ and pk, the algorithm proceeds as follows. Initialise multiset $L \leftarrow \emptyset$ and compute $s \leftarrow A_1^{\mathcal{O}_\mathcal{A}}(\mathfrak{m}, pk)$, handling any oracle calls from A_1 as follows:

- $\mathcal{O}_\mathcal{A}(v_0, v_1)$: compute $b \leftarrow \mathcal{O}_\mathcal{B}(v_0, v_1); L \leftarrow L \cup \{(b, v_0, v_1)\}; \mathfrak{bb} \leftarrow \mathsf{BB}(\mathfrak{bb}, b)$.
- $\mathcal{O}_\mathcal{A}(b)$: compute $\mathfrak{bb} \leftarrow \mathsf{BB}(\mathfrak{bb}, b)$.
- $\mathcal{O}_\mathcal{A}()$: output $\mathfrak{bb}$.

Let $L_0 \leftarrow \{v_0 \mid b \in \mathfrak{bb} \wedge (b, v_0, v_1) \in L\}$ and $L_1 \leftarrow \{v_1 \mid b \in \mathfrak{bb} \wedge (b, v_0, v_1) \in L\}$. If $L_0 = L_1$, then output $(\mathfrak{bb}, (s, L_0, L_1))$. Otherwise, compute $\mathfrak{bb}' \leftarrow \mathfrak{bb} \setminus \{b \mid b \in \mathfrak{bb} \wedge (b, v_0, v_1) \in L\}$ and output $(\mathfrak{bb}', (s, L_0, L_1))$.

The embedded adversary A_1 sees the same distibution of all elements as in the $\mathsf{IND\text{-}SEC}$ game, in particular, the simulation of $\mathcal{O}_{\mathcal{A}}()$ ensures that A_1's view of the bulletin board is consistent with $\mathsf{IND\text{-}SEC}$. The simulation of $\mathcal{O}_{\mathcal{A}}()$ also ensures that the multiset L generated by B_1 is the same as the multiset generated by $\mathcal{O}_{\mathcal{B}}$.

Algorithm B_2. Given input $\mathfrak{v}$, aux and (s, L_0, L_1), the algorithm computes g as follows:

$$g \leftarrow \begin{cases} A_2(\mathfrak{v}, aux, s) & \text{if } L_0 = L_1 \\ A_2(\emptyset, \perp, s) & \text{else if } \mathfrak{v} = \emptyset \text{, denoting failure} \\ A_2(\mathfrak{v} \cup L_0, \perp, s) & \text{otherwise} \end{cases}$$

Output g.

It is sufficient to show that the adversary $\mathcal{B}$ guesses β correctly with the same advantage as $\mathcal{A}$ in the following two cases. Case I: $L_0 = L_1$. By definition of B_1, the bulletin board $\mathfrak{bb}$ contains exactly the ballots added by $\mathcal{O}_{\mathcal{A}}(\cdot)$ and $\mathcal{O}_{\mathcal{A}}(\cdot,\cdot)$ queries. Moreover, we have $\{v_0 \mid b \in \mathfrak{bb} \wedge (b, v_0, v_1) \in L\} = \{v_1 \mid b \in \mathfrak{bb} \wedge (b, v_0, v_1) \in L\}$, as required by the challenger. It follows that the embedded adversary A_2 sees the same distibution of all elements as in $\mathsf{IND\text{-}SEC}$, hence, adversary $\mathcal{B}$ guesses β correctly with the same advantage as $\mathcal{A}$, i.e., $\mathsf{IND\text{-}SEC}^{\#}_{\mathcal{A},\Gamma}(n) \leq \mathsf{negl}(n)$. Case II: $L_0 \neq L_1$. By definition of B_1, the bulletin board $\mathfrak{bb}$ contains exactly the ballots added by $\mathcal{O}_{\mathcal{A}}(\cdot)$ queries. Since $\mathfrak{bb}$ does not contain any ballots added by $\mathcal{O}_{\mathcal{A}}(\cdot,\cdot)$ queries, we have $\emptyset = \{v_0 \mid b \in \mathfrak{bb} \wedge (b, v_0, v_1) \in L\} = \{v_1 \mid b \in \mathfrak{bb} \wedge (b, v_0, v_1) \in L\}$. Suppose $\mathfrak{bb}'$ is such that $\mathfrak{bb} = \mathfrak{bb}' \setminus \{b \mid b \in \mathfrak{bb} \wedge (b, v_0, v_1) \in L\}$, i.e., $\mathfrak{bb}'$ is the bulletin board after B_1 computed $s \leftarrow A_1^{\mathcal{O}_{\mathcal{A}}}(\mathfrak{m}, pk)$. By the correctness property of Γ, we have $(\mathfrak{v}', aux') \leftarrow \mathsf{Tally}_{sk}(\mathfrak{bb}')$ such that either: $\mathfrak{v} = \emptyset \wedge \mathfrak{v}' = \emptyset$, $\mathfrak{v} \neq \emptyset \wedge \mathfrak{v}' = \mathfrak{v} \cup L_0 \wedge \beta = 0$, or $\mathfrak{v} \neq \emptyset \wedge \mathfrak{v}' = \mathfrak{v} \cup L_1 \wedge \beta = 1$. It follows that the embedded adversary A_2 sees the same distibution of all elements as in $\mathsf{IND\text{-}SEC}$, hence, adversary $\mathcal{B}$ guesses β correctly with the same advantage as $\mathcal{A}$, i.e., $\mathsf{IND\text{-}SEC}^{\#}_{\mathcal{A},\Gamma}(n) \leq \mathsf{negl}(n)$. By Definition 3, election scheme Γ does not satisfy ballot secrecy, concluding our proof. □

References

[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.

[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.

[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.

[SB13] Ben Smyth and David Bernhard. Ballot secrecy and ballot independence coincide. In *ESORICS'13: 18th European Symposium on Research in Computer Security*, volume 8134 of *LNCS*, pages 463–480. Springer, 2013.

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