Surveying definitions of election verifiability

Ben Smyth[1,2] and Michael R. Clarkson[3]

[1]University of Birmingham, UK
[2]University of Luxembourg, Luxembourg
[3]Cornell University, Ithaca, NY, US

March 5, 2022

Abstract

We explore definitions of verifiability by Juels et al. (2010), Cortier et al. (2014), and Kiayias et al. (2015). We discover that voting systems vulnerable to attacks can be proven to satisfy each of those definitions and conclude they are unsuitable for the analysis of voting systems. Our results will fuel the exploration for a new definition.

1 Introduction

Electronic voting systems for large-scale public elections place extensive trust in software and hardware. Unfortunately, instead of being trustworthy, many are vulnerable to attacks that could unduly influence election outcomes [KSRW04, WWH+10, JS12]: Trusting voting systems is unwise; proving that systems can detect undue influence is essential.

Election verifiability enables determination of whether a voting system is vulnerable to undue influence, regardless of whether system software and hardware are trustworthy [CRS05, Adi06, Dag07, Adi08, JCJ10]. Kremer et al. [KRS10] decompose election verifiability into aspects including:

- *Individual verifiability:* voters can check that their own ballots are recorded.
- *Universal verifiability:* anyone can check that the tally of recorded ballots is computed properly.

Definitions of universal verifiability seem to originate with Benaloh and Tuinstra [BT94], who define a correctness property asserting that every participant

This manuscript has been accepted for publication in Information Processing Letters, a preliminary version appeared in a technical report [SFC15].

is convinced that the tally is accurate with respect to votes cast, and with Cohen and Fischer [CF85], who define verifiability to mean that there exists a function that accepts the announced tally if and only if the announced tally corresponds to cast votes.

Juels et al. [JCJ10, §3] define properties they name *correctness* and *verifiability* to formalize election verifiability (we rename those properties *JCJ-correctness* and *JCJ-verifiability* to avoid ambiguity), Cortier et al. [CGGI14] directly formulate definitions of individual and universal verifiability, and Kiayias et al. [KZZ15] define a property they name *E2E verifiability* (E2E abbreviates "end-to-end") to formalize individual and universal verifiability. We explore those definitions.

Contribution. We prove that definitions by Juels et al. and Cortier et al. do not detect new classes of collusion, biasing and malicious-key attacks. We also identify the definition by Kiayias et al. as not detecting some biasing attacks.

- *Collusion attacks.* A voting system's tallying and verification algorithms might be designed such that they collude to accept illegitimate tallies. Examples of collusion attacks include vote stuffing, and announcing tallies that are independent of the election.

- *Biasing attacks.* A voting system's verification algorithm might be designed to reject some legitimate tallies. Examples of biasing attacks include rejecting tallies in which a particular candidate does not win, and rejecting all tallies, even correct ones.

- *Malicious key attacks.* A voting system's verification algorithm might be designed to accept some illegitimate tallies or reject some legitimate ones, in the presence of a maliciously generated key. Examples of malicious key attacks include accepting or rejecting all tallies, regardless of their legitimacy.

In complimentary work, Smyth [Smy20] shows insecure voting systems can be proven to satisfy global verifiability, an alternative, holistic notion of verifiability. Together, these works demonstrate the need for a new verifiability definition.

Our results are presented in the context of centralised, public-key based voting systems comprising of (at least) the following four steps: First, a tallier generates a key pair $PK_{\mathcal{T}}$, $SK_{\mathcal{T}}$. Secondly, each voter constructs and casts a ballot b for their vote β. These ballots are recorded on a bulletin board BB. Thirdly, the tallier tallies the ballots recorded on the bulletin board, to derive tally X and proof P of correct tallying. Finally, voters and other interested parties verify the election to determine whether that tally corresponds to votes expressed by recorded ballots. Algorithms Setup, Vote, Tally, and Verify may be used in those steps.

2 Collusion attacks

Two examples of collusion attacks as follows:

- **Vote stuffing.** Tally behaves normally, but adds κ votes for candidate β. Verify subtracts κ votes from β, then proceeds with verification as normal. Elections thus verify as normal, except that candidate β receives extra votes.

- **Backdoor tally replacement.** Tally and Verify behave normally, unless a *backdoor* value is posted on the bulletin board. For example, if $(SK_{\mathcal{T}}, X^*)$ appears on the board, then Tally and Verify both ignore the correct tally and instead replace it with tally X^*. The tallier's private key $SK_{\mathcal{T}}$ is the backdoor here, it cannot appear on the bulletin board (except with negligible probability) unless the tallier is malicious.

Intuitively, vote stuffing and backdoor tally replacement attacks should be detected by universal verifiability, because that notion should require Verify to accept only those tallies that correspond to a correct tally of the bulletin board. (Voting stuffing attacks should also be detected by correctness, because it should require the tally produced by Tally to correspond to the votes encapsulated in ballots on the bulletin board.)

Elections are big business; history teaches us that malice is rife. Talliers can be incentivized to act in their own selfish interests, rather than serve democracy. They should never be trusted—that's why verifiability emerged as an essential property of voting systems. Accordingly, we should anticipate talliers launching backdoor tally replacement attacks, even though a honest tallier would never relinquish their private key.

Design guideline (Soundness)**.** *Verification must only accept tallies that correspond to votes expressed in collected ballots.*

We formalize a vote stuffing attack by modifying an election scheme Π to derive a vote-stuffing election scheme $\mathsf{Stuff}(\Pi, \beta, \kappa)$, which adds κ votes for candidate β, wherein the modified scheme:

- Tallies ballots using Π's tallying algorithm to compute tally X and a proof, and announces tally $\mathtt{Add}(X, \beta, \kappa)$ along with that proof.

- Verifies elections using Π's verification algorithm with respect to tally $\mathtt{Add}(X, \beta, -\kappa)$.

where $\mathtt{Add}(X, \beta, \kappa) = (X[1], \ldots, X[\beta - 1], X[\beta] + \kappa, X[\beta + 1], \ldots, X[|X|])$ and $|X|$ denotes the length of vector X. (Note that Juels et al. define tally X as a vector such that $X[j]$ is the number of votes for the jth candidate.) The definition of election verifiability by Juels et al. does not detect vote stuffing attacks:

Proposition 1. *Given an election scheme* Π *satisfying JCJ-correctness and JCJ-verifiability, and integers* β *and* κ*, it holds that* $\mathsf{Stuff}(\Pi, \beta, \kappa)$ *satisfies JCJ-correctness and JCJ-verifiability.*

We state definitions by Juels et al. for an election scheme, JCJ-correctness, and JCJ-verifiability, along with a proof of Proposition 1 in Section 5.

We formalize a backdoor tally replacement attack by modifying an election scheme Π to derive an election scheme $\mathsf{Backdoor}(\Pi)$, where the modified scheme:

- Tallies ballots as follows: If there exists a unique X^* such that $(SK_{\mathcal{T}}, X^*) \in BB$, then announce tally X^* and some arbitrary proof $\perp$, otherwise, announce a tally and proof computed using Π's tallying algorithm.
- Verification proceeds as follows: If there exists a unique X^* such that $(SK_{\mathcal{T}}, X^*) \in BB$,[1] then accept, otherwise, verify the election using Π's verification algorithm.

The definition of election verifiability by Juels et al. does not detect backdoor tally replacement attacks:

Proposition 2. *Given an election scheme* Π *satisfying JCJ-correctness and JCJ-verifiability, it holds that* $\mathsf{Backdoor}(\Pi)$ *satisfies JCJ-correctness and JCJ-verifiability, assuming the scheme does not leak the tallier's private key.*

A proof appears in Section 5, where we also define key leakage.

Cortier et al. propose a definition similar to JCJ-verifiability and insist that election schemes must satisfy their notions of correctness and partial tallying. Vote stuffing attacks are detected by their notion of correctness, and $\mathsf{Backdoor}(\Pi)$ does not satisfy their notion of partial tallying. Nonetheless, we formalize a further backdoor tally replacement attack by modifying $\mathsf{Backdoor}$ to derive $\mathsf{Backdoor}'$, which additionally checks whether the most significant bit of private key $SK_{\mathcal{T}}$ is zero. The definition of election verifiability by Cortier et al. does not detect that backdoor tally replacement attack:

Proposition 3. *Given an election scheme* $\mathbb{V}$*,* $\mathbb{R}$*,* ρ*,* Π *satisfying Cortier et al. verifiability, it holds that* $\mathbb{V}$*,* $\mathbb{R}$*,* ρ*,* $\mathsf{Backdoor}'(\Pi)$ *satisfies Cortier et al. verifiability, assuming the scheme does not leak the tallier's private key and the most significant bit of an honestly generated private key is one.*

We state definitions by Cortier et al. for an election scheme and verifiability, along with a proof of Proposition 3 in Section 6, where we also define key leakage for such schemes.

[1] The pre-condition must also check that $SK_{\mathcal{T}}$ is the private key corresponding to public key $PK_{\mathcal{T}}$. We omit formalizing this detail, noting it is straightforward for encryption schemes such as El Gamal and RSA.

3 Biasing attacks

We derive the following three examples of biasing attacks from an election scheme Π, by modifying verification:

- **Reject all.** Election scheme $\mathsf{Reject}(\Pi)$ always rejects; no election will ever be considered valid.
- **Selective reject.** Let ε be a distinguished value that would not be posted on the bulletin board by honest voters. Election scheme $\mathsf{Selective}(\Pi, \varepsilon)$ defines verification as follows: If Π's verification algorithm accepts and $\varepsilon \notin BB$, then accept, otherwise, reject. Since elections are invalidated when ε appears on the bulletin board, some elections can be nullified.
- **Biased reject.** Let Z be a set of tallies. Election scheme $\mathsf{Bias}(\Pi, Z)$ defines verification as follows: If Π's verification algorithm accepts a tally in Z, then accept, otherwise, reject. Elections are biased toward tallies in Z, because verification only accepts such tallies.

Intuitively, these formalizations should not satisfy universal verifiability, because that notion should require Verify to always accept tallies that correspond to a correct tally of the bulletin board.

Design guideline (Completeness)**.** *Verification must always accept tallies that correspond to votes expressed in collected ballots.*

The definition of verifiability by Juels et al. does not detect any of the above three attacks, because that definition has no notion of completeness. For example, it is vulnerable to biased reject attacks:

Proposition 4. *Given an election scheme Π satisfying JCJ-correctness and JCJ-verifiability, and given a set Z, it holds that $\mathsf{Bias}(\Pi, Z)$ satisfies JCJ-correctness and JCJ-verifiability.*

A proof sketch appears in Section 5.

The definition of verifiability by Cortier et al. detects biased reject and reject all attacks, but does not detect selective reject attacks, because that definition's notion of completeness does not quantify over all bulletin boards:

Proposition 5. *Given an election scheme $\mathbb{V}$, $\mathbb{R}$, ρ, Π satisfying Cortier et al. verifiability and symbol ε that does not appear in the co-domain of Vote, it holds that $\mathbb{V}$, $\mathbb{R}$, ρ, $\mathsf{Selective}(\Pi, \varepsilon)$ satisfies Cortier et al. verifiability.*

A proof sketch appears in Section 6.

The definition of verifiability by Kiayias et al. does not detect selective reject attacks either, because (like Juels et al.) the definition has no notion of completeness. Their notion of correctness rules out reject all and biased reject attacks. The ideas remain the same, so we omit formalized results.

4 Malicious key attacks

We derive the following two examples of malicious key attacks from an election scheme Π, for which the most significant bit of honestly generated public keys is one:

- **Malicious key accept.** Let $\mathsf{Accept}(\Pi)$ be Π except, firstly, verification is defined as follows: If the most significant bit of the tallier's public key is zero, then accept, otherwise, verify using Π's verification algorithm. Secondly, other algorithms are the same except they remove the most significant bit of public keys when that bit is zero. Thus, elections verify as normal, except when a tallier announces a public key with a maliciously prepended zero.

- **Malicious key reject.** Let Reject be Accept, except verification rejects when the most significant bit is zero.

Intuitively, these formalizations should not satisfy universal verifiability, because soundness and completeness should, respectively, preclude the former and latter class of attacks. Yet, the definitions of verifiability by Juels et al. and Cortier et al. do not detect the above attacks, because their definitions do not consider maliciously generated keys:

Proposition 6. *Given an election scheme* Π *satisfying JCJ-correctness and JCJ-verifiability, it holds that both* $\mathsf{Accept}(\Pi)$ *and* $\mathsf{Reject}(\Pi)$ *satisfy JCJ-correctness and JCJ-verifiability, assuming the most significant bit of honestly generated public keys is one.*

Proposition 7. *Given an election scheme* $\mathbb{V}$*,* $\mathbb{R}$*,* ρ*,* Π *satisfying Cortier et al. verifiability, it holds that both* $\mathbb{V}$*,* $\mathbb{R}$*,* ρ*,* $\mathsf{Accept}(\Pi)$ *and* $\mathbb{V}$*,* $\mathbb{R}$*,* ρ*,* $\mathsf{Reject}(\Pi)$ *satisfy Cortier et al. verifiability, assuming the most significant bit of honestly generated public keys is one.*

Proofs follow immediately from the definitions of verifiability by Juels et al. and Cortier et al., because those definitions consider only honestly generated keys: The former explicitly assumes keys pairs are generated by "a trusted third party [or] on an interactive computationally secure key-generation protocol." The latter generates key pairs using algorithm Setup (cf. the opening lines of games in Definition 4). Maliciously generated keys are not considered.

5 Juels et al. definitions and related proofs

We state formal definitions of election schemes and verifiability by Juels et al., and prove that their security definition does not detect collusion nor biasing attacks.

5.1 Syntax

In addition to a tallier and voters, a registrar in possession of key pair $PK_{\mathcal{R}}, SK_{\mathcal{R}}$ is considered. That registrar generates voter-credential pairs pk, sk, before ballots are cast. Election schemes are defined as tuples $(\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ of probabilistic polynomial-time algorithms:[2]

- **Register**, denoted $(pk, sk) \leftarrow \mathsf{Register}(SK_{\mathcal{R}}, i, k_1)$, is executed by the registrar. Register takes as input the private key $SK_{\mathcal{R}}$ of the registrar, a voter's identity i, and security parameter k_1. It outputs a credential pair (pk, sk).

- **Vote**, denoted $b \leftarrow \mathsf{Vote}(sk, PK_{\mathcal{T}}, n_C, \beta, k_2)$, is executed by voters. Vote takes as input a voter's private credential sk, the public key $PK_{\mathcal{T}}$ of the tallier, the number of candidates n_C, the voter's vote β, and security parameter k_2. It outputs a ballot b.

- **Tally**, denoted $(\mathbf{X}, P) \leftarrow \mathsf{Tally}(SK_{\mathcal{T}}, BB, n_C, \{pk_i\}_{i=1}^{n_V}, k_3)$, is executed by the tallier. Tally takes as input the private key $SK_{\mathcal{T}}$ of the tallier, the bulletin board BB, the number of candidates n_C, the set containing voters' public credentials, and security parameter k_3. It outputs the tally $\mathbf{X}$ and a proof P that the tally is correct, where $\mathbf{X}$ is a vector of length n_C such that $\mathbf{X}[j]$ indicates the number of votes for the jth candidate.

- **Verify**, denoted $v \leftarrow \mathsf{Verify}(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{X}, P)$, can be executed by anyone to verify the election. Verify takes as input the public key $PK_{\mathcal{R}}$ of the registrar, the public key $PK_{\mathcal{T}}$ of the tallier, the bulletin board BB, the number of candidates n_C, and a candidate proof P of correct tallying. It outputs a bit v, which is 1 if the tally successfully verifies and 0 on failure.

The above syntax fixes an apparent oversight in the original presentation: we supply the registrar's public key as input to the verification algorithm, because that key would be required by Verify to check the signature on the electoral roll.

5.2 Security definitions

Juels et al. formalize *correctness* and *verifiability* to capture their notion of election verifiability. We rename those to *JCJ-correctness* and *JCJ-verifiability* to avoid ambiguity. For readability, the definitions we give below contain subtle differences from the original presentation. For example, we sometimes use for loops instead of pattern matching.

JCJ-correctness asserts that an adversary cannot modify or eliminate votes of honest voters, and stipulates that at most one ballot is tallied per voter. Intuitively, the security definition challenges the adversary to ensure that verification succeeds and the tally does not include some honest votes or contains

[2]Juels et al. do not explicitly name key generation algorithms. For consistency, readers may like to consider Setup generating tallier key pairs.

too many votes. Our presentation of JCJ-correctness fixes apparent errors in the original: the adversary is given the credentials for corrupt voters and distinct security parameters are supplied to the $\mathsf{Register}$ and Vote algorithms. An implicit assumption is also omitted: $\{\beta_i\}_{i\in\mathcal{V}\setminus\mathcal{V}'}$ is a multiset of valid votes, that is, for all $\beta \in \{\beta_i\}_{i\in\mathcal{V}\setminus\mathcal{V}'}$ we have $1 \leq \beta \leq n_C$. Without this assumption the security definition cannot be satisfied by many election schemes, including the election scheme by Juels et al.

Definition 1 (JCJ-correctness)**.** *An election scheme* $\Pi = (\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ *satisfies* JCJ-correctness *if for all probabilistic polynomial-time adversary* $\mathcal{A}$*, there exists a negligible function* μ*, such that for all positive integers* n_C *and* n_V*, and security parameters* k_1*,* k_2*, and* k_3*, we have* $\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Cor}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3)) \leq \mu(k_1, k_2, k_3)$,[3] *where:*

$\mathsf{Exp\text{-}JCJ\text{-}Cor}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3) =$

$\mathcal{V} \leftarrow \{1, \ldots, n_V\}$;
for $i \in \mathcal{V}$ **do** $(pk_i, sk_i) \leftarrow \mathsf{Register}(SK_{\mathcal{R}}, i, k_1)$;
$\mathcal{V}' \leftarrow \mathcal{A}(\{pk_i\}_{i=1}^{n_V})$;
for $i \in \mathcal{V} \setminus \mathcal{V}'$ **do** $\beta_i \leftarrow \mathcal{A}()$;
$BB \leftarrow \{\mathsf{Vote}(sk_i, PK_{\mathcal{T}}, n_C, \beta_i, k_2)\}_{i\in\mathcal{V}\setminus\mathcal{V}'}$;
$(\mathbf{X}, P) \leftarrow \mathsf{Tally}(SK_{\mathcal{T}}, BB, n_C, \{pk_i\}_{i=1}^{n_V}, k_3)$;
$BB \leftarrow BB \cup \mathcal{A}(BB, \{(pk_i, sk_i)\}_{i\in\mathcal{V}\cap\mathcal{V}'})$;
$(\mathbf{X}', P') \leftarrow \mathsf{Tally}(SK_{\mathcal{T}}, BB, n_C, \{pk_i\}_{i=1}^{n_V}, k_3)$;
if $\mathsf{Verify}(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{X}', P') = 1$
$\wedge \big(\{\beta_i\}_{i\in\mathcal{V}\setminus\mathcal{V}'} \not\subseteq \langle\mathbf{X}'\rangle \vee |\langle\mathbf{X}'\rangle| - |\langle\mathbf{X}\rangle| > |\mathcal{V}'|\big)$ **then**
  return 1;
else
  return 0;

and $\langle\mathbf{X}\rangle$ *denotes the translation of tally* $\mathbf{X}$ *to multiset* $\bigcup_{1\leq j\leq|\mathbf{X}|}\{\underbrace{j, \ldots, j}_{\mathbf{X}[j]\ times}\}$.

The JCJ-correctness definition implicitly assumes that the tally and associated proof are honestly computed using algorithm Tally. By comparison, the definition of JCJ-verifiability does not use this assumption, hence, JCJ-verifiability is intended to assert that voters and auditors can check whether votes have been recorded and tallied correctly. Intuitively, the adversary is assumed to control the tallier and voters, and the security definition challenges the adversary to concoct an election (that is, the adversary generates a bulletin board BB, a tally $\mathbf{X}$, and a proof of tallying P) such that verification succeeds and tally $\mathbf{X}$ differs tally $\mathbf{X}'$ derived from honestly tallying the bulletin board BB. It follows that there is at most one verifiable tally that can be derived.

Definition 2 (JCJ-verifiability)**.** *An election scheme* $\Pi = (\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ *satisfies* JCJ-verifiability *if for all probabilistic polynomial-time adversary* $\mathcal{A}$*, there exists a negligible function* μ*, such that for all positive integers* n_C

[3] We write $\mu(k_1, k_2, k_3)$ for the smallest value in $\{\mu(k_1), \mu(k_2), \mu(k_3)\}$ (cf. [JCJ10, pp45]).

and n_V*, and security parameters* k_1 *and* k_3*, we have* $\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Ver}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3)) \leq \mu(k_1, k_2, k_3)$*, where:*

$\mathsf{Exp\text{-}JCJ\text{-}Ver}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3) =$

for $1 \leq i \leq n_V$ **do** $(pk_i, sk_i) \leftarrow \mathsf{Register}(SK_{\mathcal{R}}, i, k_1)$;
$(BB, \mathbf{X}, P) \leftarrow \mathcal{A}(SK_{\mathcal{T}}, \{(pk_i, sk_i)\}_{i=1}^{n_V})$;
$(\mathbf{X}', P') \leftarrow \mathsf{Tally}(SK_{\mathcal{T}}, BB, n_C, \{pk_i\}_{i=1}^{n_V}, k_3)$;
if $\mathsf{Verify}(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{X}, P) = 1 \wedge \mathbf{X} \neq \mathbf{X}'$ **then**
  return 1;
else
  return 0;

5.3 Proof of Proposition 1

Suppose $\Pi = (\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ is an election scheme satisfying JCJ-correctness and JCJ-verifiability. Further suppose $\mathsf{Stuff}(\Pi, \beta, \kappa) = (\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}_S, \mathsf{Verify}_S)$, for some integers $\beta, \kappa \in \mathbb{N}$. We prove that $\mathsf{Stuff}(\Pi, \beta, \kappa)$ satisfies JCJ-correctness and JCJ-verifiability.

We show that $\mathsf{Stuff}(\Pi, \beta, \kappa)$ satisfies JCJ-correctness by contradiction. Suppose $\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Cor}(\mathsf{Stuff}(\Pi, \beta, \kappa), \mathcal{A}, n_C, n_V, k_1, k_2, k_3))$ is non-negligible for some k_1, k_2, k_3, n_C, n_V, and $\mathcal{A}$. Hence, there exists an execution of the experiment $\mathsf{Exp\text{-}JCJ\text{-}Cor}(\mathsf{Stuff}(\Pi, \beta, \kappa), \mathcal{A}, n_C, n_V, k_1, k_2, k_3)$ that satisfies

$$\mathsf{Verify}_S(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{X}', P') = 1 \\ \wedge \left(\{\beta_i\}_{i \in \mathcal{V} \setminus \mathcal{V}'} \not\subseteq \langle \mathbf{X}' \rangle \vee |\langle \mathbf{X}' \rangle| - |\langle \mathbf{X} \rangle| > |\mathcal{V}'|\right)$$

with non-negligible probability, where $\{\beta_i\}_{i \in \mathcal{V} \setminus \mathcal{V}'}$ is the set of honest votes, $(\mathbf{X}, P)$ is the tally of honest votes, $(\mathbf{X}', P')$ is the tally of all votes, $\mathcal{V}'$ is a set of corrupt voter identities, and BB is the bulletin board. Further suppose BB_0 is the bulletin board BB before adding adversarial ballots. By definition of Tally_S, there exist computations

$$(\mathbf{Y}, Q) \leftarrow \mathsf{Tally}(SK_{\mathcal{T}}, BB_0, n_C, \{pk_i\}_{i=1}^{n_V}, k_3)$$

and

$$(\mathbf{Y}', Q') \leftarrow \mathsf{Tally}(SK_{\mathcal{T}}, BB, n_C, \{pk_i\}_{i=1}^{n_V}, k_3)$$

such that $\mathbf{X} = \mathtt{Add}(\mathbf{Y}, \beta, \kappa)$, $\mathbf{X}' = \mathtt{Add}(\mathbf{Y}', \beta, \kappa)$, and $P' = Q'$. Since $\kappa \in \mathbb{N}$, we have $\langle \mathbf{Y}' \rangle \subseteq \langle \mathbf{X}' \rangle$. Moreover, $|\langle \mathbf{X} \rangle| = |\langle \mathbf{Y} \rangle| + \kappa$ and $|\langle \mathbf{X}' \rangle| = |\langle \mathbf{Y}' \rangle| + \kappa$, hence,

$$|\langle \mathbf{Y}' \rangle| - |\langle \mathbf{Y} \rangle| = |\langle \mathbf{X}' \rangle| - |\langle \mathbf{X} \rangle|.$$

By definition of Verify_S and since $\mathbf{Y}' = \mathtt{Add}(\mathbf{X}', \beta, -\kappa)$, there exists a computation

$$v \leftarrow \mathsf{Verify}_0(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{Y}', Q')$$

such that $v = 1$. It follows that

$$\mathsf{Verify}(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{Y}', Q') = 1 \\ \wedge \left(\{\beta_i\}_{i\in\mathcal{V}\setminus\mathcal{V}'} \not\subseteq \langle \mathbf{Y}' \rangle \vee |\langle \mathbf{Y}' \rangle| - |\langle \mathbf{Y} \rangle| > |\mathcal{V}'|\right)$$

with non-negligible probability and, furthermore, we have $\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Cor}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3))$ is non-negligible, thereby deriving a contradiction.

We show that $\mathsf{Stuff}(\Pi, \beta, \kappa)$ satisfies JCJ-verifiability by contradiction. Suppose $\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Ver}(\mathsf{Stuff}(\Pi, \beta, \kappa), \mathcal{A}, n_C, n_V, k_1, k_2, k_3))$ is non-negligible for some k_1, k_3, n_C, n_V, and $\mathcal{A}$. Hence, there exists an execution of the experiment $\mathsf{Exp\text{-}JCJ\text{-}Ver}(\mathsf{Stuff}(\Pi, \beta, \kappa), \mathcal{A}, n_C, n_V, k_1, k_2, k_3)$ which satisfies

$$\mathsf{Verify}(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{X}, P) = 1 \wedge \mathbf{X} \neq \mathbf{X}'$$

with non-negligible probability, where $(BB, \mathbf{X}, P)$ is an election concocted by the adversary and $(\mathbf{X}', P')$ is produced by tallying BB. By definition of Tally_S, there exists a computation

$$(\mathbf{Y}', Q') \leftarrow \mathsf{Tally}(SK_{\mathcal{T}}, BB, n_C, \{pk_i\}_{i=1}^{n_V}, k_3)$$

such that $\mathbf{X}' = \mathtt{Add}(\mathbf{Y}', \beta, \kappa)$ and $P' = Q'$. By definition of Verify_S, there exists a computation

$$v \leftarrow \mathsf{Verify}(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathtt{Add}(\mathbf{X}, \beta, -\kappa), P)$$

such that $v = 1$. Let the adversary $\mathcal{B}$ be defined as follows: given input K and S, the adversary $\mathcal{B}$ computes

$$(BB, \mathbf{X}, P) \leftarrow \mathcal{A}(K, S)$$

and outputs $(BB, \mathtt{Add}(\mathbf{X}, \beta, -\kappa), P)$. We have an execution of the experiment $\mathsf{Exp\text{-}JCJ\text{-}Ver}(\mathsf{Stuff}(\Pi, \beta, \kappa), \mathcal{B}, n_C, n_V, k_1, k_2, k_3)$ that concocts the election $(BB, \mathtt{Add}(\mathbf{X}, \beta, -\kappa), P)$ and tallying BB produces $(\mathbf{Y}', Q')$ such that

$$\mathsf{Verify}(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathtt{Add}(\mathbf{X}, \beta, -\kappa), P) = 1$$

with non-negligible probability. Moreover, since $\mathbf{X} \neq \mathbf{X}'$ and $\mathbf{Y}' = \mathtt{Add}(\mathbf{X}', \beta, -\kappa)$, we have $\mathtt{Add}(\mathbf{X}, \beta, -\kappa) \neq \mathbf{Y}'$ with non-negligible probability. It follows immediately that $\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Cor}(\Pi, \mathcal{B}, n_C, n_V, k_1, k_2, k_3))$ is non-negligible, thus deriving a contradiction and concluding our proof. $\square$

5.4 Proof of Proposition 2

We define key leakage before proving Proposition 2.

Definition 3 (Key leakage)**.** *An election scheme* $\Pi = (\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ does not leak the tallier's private key *if for all positive integers n_C and n_V, security parameters k_1 and k_3, and probabilistic polynomial-time adversary $\mathcal{A}$, we have* $\mathsf{Succ}(\mathsf{Exp\text{-}leak}(\Pi, \mathcal{A}, k_1, k_3, n_C, n_V))$ *is negligible, where:*

$\mathsf{Exp\text{-}leak}(\Pi, \mathcal{A}, k_1, k_3, n_C, n_V) =$

for $1 \leq i \leq n_V$ **do** $(pk_i, sk_i) \leftarrow \mathsf{Register}(SK_{\mathcal{R}}, i, k_1)$;
$BB \leftarrow \mathcal{A}(sk_1, \ldots, sk_{n_V})$;
$(\mathbf{X}, P) \leftarrow \mathsf{Tally}(SK_{\mathcal{T}}, BB, n_C, \{pk_i\}_{i=1}^{n_V}, k_3)$;
if $\mathcal{A}(\mathbf{X}, P) = SK_{\mathcal{T}}$ **then**
  return 1;
else
  return 0;

Proof of Proposition 2. Suppose $\Pi = (\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ is an election scheme satisfying JCJ-correctness and JCJ-verifiability that does not leak the tallier's private key, and$\mathsf{Backdoor}(\Pi) = (\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}_B, \mathsf{Verify}_B)$. Since Π does not leak the tallier's private key, the experiment $\mathsf{Exp\text{-}JCJ\text{-}Cor}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3)$ produces a bulletin board BB such that $SK_{\mathcal{T}} \in BB$ with negligible probability, for all k_1, k_2, k_3, n_C, n_V and $\mathcal{A}$. It follows that an execution of the experiment $\mathsf{Exp\text{-}JCJ\text{-}Cor}(\mathsf{Backdoor}(\Pi), \mathcal{A}, n_C, n_V, k_1, k_2, k_3)$ is an execution of the experiment $\mathsf{Exp\text{-}JCJ\text{-}Cor}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3)$ with non-negligible probability and, hence, $\mathsf{Backdoor}(\Pi)$ satisfies JCJ-correctness.

We show that $\mathsf{Backdoor}(\Pi)$ satisfies JCJ-verifiability by contradiction. Suppose $\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Ver}(\mathsf{Backdoor}(\Pi), \mathcal{A}, n_C, n_V, k_1, k_2, k_3))$ is non-negligible for some k_1, k_3, n_C, n_V, and $\mathcal{A}$. Hence, there exists an execution of the experiment $\mathsf{Exp\text{-}JCJ\text{-}Ver}(\mathsf{Backdoor}(\Pi), \mathcal{A}, n_C, n_V, k_1, k_2, k_3)$ which satisfies

$$\mathsf{Verify}(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{X}, P) = 1 \land \mathbf{X} \neq \mathbf{X}'$$

with non-negligible probability, where $(BB, \mathbf{X}, P)$ is an election concocted by the adversary and $(\mathbf{X}', P')$ is produced by tallying BB. If there is one and only one $\mathbf{Y}$ such that $(SK_{\mathcal{T}}, \mathbf{Y}) \in BB$, then $\mathbf{X}' = \mathbf{Y}$ by definition of Tally and $\mathbf{X} = \mathbf{Y}$ by definition of Verify, otherwise, the execution of the experiment $\mathsf{Exp\text{-}JCJ\text{-}Cor}(\mathsf{Backdoor}(\Pi), \mathcal{A}, n_C, n_V, k_1, k_2, k_3)$ is an execution of the experiment $\mathsf{Exp\text{-}JCJ\text{-}Cor}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3)$ and, hence,

$$\begin{aligned}&\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Ver}(\mathsf{Backdoor}(\Pi), \mathcal{A}, n_C, n_V, k_1, k_2, k_3)) \\ &\qquad = \mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Ver}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3)).\end{aligned}$$

In both cases we derive a contradiction, thereby concluding our proof. □

5.5 Proof sketch of Proposition 4

Suppose $\Pi = (\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify})$ is an election scheme satisfying JCJ-correctness and JCJ-verifiability. Further suppose $\mathsf{Bias}(\Pi, Z) = (\mathsf{Register}, \mathsf{Vote}, \mathsf{Tally}, \mathsf{Verify}_R)$, for some set of vectors Z. By definition of Verify_R, we have

$$\mathsf{Verify}_R(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{X}, P) = 1$$

implies the existence of a computation

$$v \leftarrow \mathsf{Verify}(PK_{\mathcal{R}}, PK_{\mathcal{T}}, BB, n_C, \mathbf{X}, P)$$

such that $v = 1$ with non-negligible probability, for all $PK_{\mathcal{T}}$, BB, n_C, $\mathbf{X}$, and P. It follows that

$$\begin{aligned}&\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Cor}(\mathsf{Bias}(\Pi), \mathcal{A}, n_C, n_V, k_1, k_2, k_3))\\ &\qquad\leq \mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Cor}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3))\end{aligned}$$

and

$$\begin{aligned}&\mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Ver}(\mathsf{Bias}(\Pi), \mathcal{A}, n_C, n_V, k_1, k_2, k_3))\\ &\qquad\leq \mathsf{Succ}(\mathsf{Exp\text{-}JCJ\text{-}Ver}(\Pi, \mathcal{A}, n_C, n_V, k_1, k_2, k_3))\end{aligned}$$

for all k_1, k_2, k_3, n_C, n_V, and $\mathcal{A}$. Hence, $\mathsf{Bias}(\Pi, Z)$ satisfies JCJ-correctness and JCJ-verifiability. □

6 Cortier et al. definitions and related proofs

We state definitions of election schemes and verifiability by Cortier et al., and prove their security definition does not detect selective reject nor backdoor tally replacement attacks.

6.1 Syntax

Election schemes are defined over a vote space $\mathbb{V}$, a result space $\mathbb{R}$, a result function $\rho : \mathbb{V} \times \cdots \times \mathbb{V} \to \mathbb{R}$, and a tuple ($\mathsf{Setup}, \mathsf{Register}, \mathsf{Vote}, \mathsf{Validate}, \mathsf{Box}, \mathsf{VerifyVote}, \mathsf{Tally}, \mathsf{Verify}$) of probabilistic polynomial-time algorithms:

- **Setup**, denoted $(PK_{\mathcal{T}}, SK_{\mathcal{T}}) \leftarrow \mathsf{Setup}(k)$, is executed by the tallier. Setup takes a security parameter k as input and outputs a key pair $(PK_{\mathcal{T}}, SK_{\mathcal{T}})$.

- **Register**, denoted $(pk, sk) \leftarrow \mathsf{Register}(PK_{\mathcal{T}}, i, k)$, is executed by the registrar. $\mathsf{Register}$ takes as input the tallier's public key $PK_{\mathcal{T}}$, a voter's identity i, and security parameter k. It outputs a credential pair (pk, sk).

- **Vote**, denoted $b \leftarrow \mathsf{Vote}(i, pk, sk, PK_{\mathcal{T}}, \beta)$, is executed by voters. Vote takes as input a voter's identity i and credential pair (pk, sk), tallier's public key $PK_{\mathcal{T}}$, and vote $\beta \in \mathbb{V}$. It outputs a ballot b.

- **Validate**, denoted $v \leftarrow \mathsf{Validate}(PK_{\mathcal{T}}, b)$, is executed by the bulletin board. $\mathsf{Validate}$ takes as input the tallier's public key $PK_{\mathcal{T}}$ and ballot b. It outputs a bit v, which is 1 for a well-formed ballot and 0 otherwise.

- **Box**, denoted $BB' \leftarrow \mathsf{Box}(PK_{\mathcal{T}}, BB, b)$, is executed by the bulletin board. Box takes as input the tallier's public key $PK_{\mathcal{T}}$, bulletin board BB, and ballot b. It outputs an updated bulletin board BB'.

- **VerifyVote**, denoted $v \leftarrow \mathsf{VerifyVote}(i, pk, sk, PK_{\mathcal{T}}, BB, b)$, is executed by voters. $\mathsf{VerifyVote}$ takes as input a voter's identity i and credential pair (pk, sk), tallier's public key $PK_{\mathcal{T}}$, bulletin board BB, and ballot b. It outputs a bit v, which is 1 if ballot b has been recorded by bulletin board BB and 0 otherwise.

- **Tally**, denoted $(X, P) \leftarrow \mathsf{Tally}(PK_{\mathcal{T}}, SK_{\mathcal{T}}, BB)$, is executed by the tallier. Tally takes as input the tallier's key pair $(PK_{\mathcal{T}}, SK_{\mathcal{T}})$ and the bulletin board BB. It outputs the tally X and a proof P that the tally is correct.

- **Verify**, denoted $v \leftarrow \mathsf{Verify}(PK_{\mathcal{T}}, BB, X, P)$, can be executed by anyone to verify the election. Verify takes as input the tallier's public key $PK_{\mathcal{T}}$, the bulletin board BB, a tally X, and a proof P of correct tallying. It outputs a bit v, which is 1 if the tally successfully verifies and 0 on failure.

Election schemes must satisfy the following properties:

Cortier et al. correctness. For all security parameters k, integers n_V and votes $\beta_1, \ldots, \beta_{n_V} \in \mathbb{V}$, we have:

$$
\begin{aligned}
Pr\Big[&(PK_{\mathcal{T}}, SK_{\mathcal{T}}) \leftarrow \mathsf{Setup}(k);\\
&\textbf{for } 1 \leq i \leq n_V \textbf{ do}\\
&\quad (pk_i, sk_i) \leftarrow \mathsf{Register}(PK_{\mathcal{T}}, i, k);\\
&\quad b_i \leftarrow \mathsf{Vote}(i, pk_i, sk_i, PK_{\mathcal{T}}, \beta_i);\\
&BB \leftarrow \{b_1, \ldots, b_{n_V}\};\\
&(X, P) \leftarrow \mathsf{Tally}(PK_{\mathcal{T}}, SK_{\mathcal{T}}, BB) :\\
&X = \rho(\beta_1, \ldots, \beta_{n_V}) \wedge \mathsf{Verify}(PK_{\mathcal{T}}, BB, X, P) = 1 \wedge\\
&\Big(\textstyle\bigwedge_{1 \leq i \leq n_V} \mathsf{Validate}(PK_{\mathcal{T}}, b_i) = 1 \wedge \mathsf{VerifyVote}(i, pk_i, sk_i, PK_{\mathcal{T}}, \{b_i\}, b_i) =\\
&1 \wedge \mathsf{Box}(PK_{\mathcal{T}}, \emptyset, b_i) = \{b_i\}\Big)\Big] = 1.
\end{aligned}
$$

Cortier et al. partial tallying. There exists a commutative binary operator $\star : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that for all integers j and k, and votes $\alpha_1, \ldots, \alpha_j, \beta_1, \ldots, \beta_k \in \mathbb{V}$, we have $\rho(\alpha_1, \ldots, \alpha_j, \beta_1, \ldots, \beta_k) = \rho(\alpha_1, \ldots, \alpha_j) \star \rho(\beta_1, \ldots, \beta_k)$. Moreover, for all security parameters k and disjoint bulletin boards BB_1 and BB_2, there exists a negligible function μ such that

$$
\begin{aligned}
\Pr[&(PK_{\mathcal{T}}, SK_{\mathcal{T}}) \leftarrow \mathsf{Setup}(k);\\
&(X_1, P_1) \leftarrow \mathsf{Tally}(PK_{\mathcal{T}}, SK_{\mathcal{T}}, BB_1);\\
&(X_2, P_2) \leftarrow \mathsf{Tally}(PK_{\mathcal{T}}, SK_{\mathcal{T}}, BB_2);\\
&(X, P) \leftarrow \mathsf{Tally}(PK_{\mathcal{T}}, SK_{\mathcal{T}}, BB_1 \cup BB_2) :\\
&X \neq \perp \Rightarrow X = X_1 \star X_2] > 1 - \mu(k).
\end{aligned}
$$

For consistency and readability, our presentation of the above syntax and properties contain some minor differences from the original presentation. In addition, we fix an apparent oversight: we define the tallier's public key and voter credentials.

6.2 Security definition

The definition of verifiability by Cortier et al. is captured by experiments similar to Exp-JCJ-Ver (Section 5.2).

Definition 4 (Cortier et al. verifiability)**.** *An election scheme* $\mathbb{V}$, $\mathbb{R}$, ρ, Π *satisfies* Cortier et al. verifiability *if for all security parameters* k *and probabilistic polynomial-time adversaries* $\mathcal{A}$, *there exists a negligible function* μ *such that* $\mathsf{Succ}(\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V},\rho,\Pi,\star,\mathcal{A},k)) + \mathsf{Succ}(\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}g}(\mathbb{V},\rho,\Pi,\star,\mathcal{A},k)) \leq \mu(k)$, *where* $\star$ *is a commutative binary operator satisfying the partial tallying property and the aforementioned experiments are defined as follows:*[4]

$\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V},\rho,\Pi,\star,\mathcal{A},k) =$

$(PK_{\mathcal{T}}, SK_{\mathcal{T}}) \leftarrow \mathsf{Setup}(k)$;
$\mathit{Crpt} \leftarrow \emptyset$; $\mathit{Reg} \leftarrow \emptyset$; $\mathit{Rvld} \leftarrow \emptyset$;
$(BB, X, P) \leftarrow \mathcal{A}^{C,E,R}(PK_{\mathcal{T}})$;
if $\mathsf{Verify}(PK_{\mathcal{T}}, BB, X, P) = 0$ **then return** 0;
if $X = \perp$ **then return** 0;
if $\exists (i_1^A, \beta_1^A, b_1^A), \ldots, (i_{n_A}^A, \beta_{n_A}^A, b_{n_A}^A) \in \mathit{Rvld} \setminus \mathit{Chck}$
$\wedge\ \exists \beta_1^B, \ldots, \beta_{n_B}^B \,.\, 0 \leq n_B \leq |\mathit{Crpt}|$
$\wedge\ X = \rho(\{\beta \mid (i,\beta,b) \in \mathit{Chck}\}) \star \rho(\{\beta_i^A\}_{i=1}^{n_A}) \star \rho(\{\beta_i^B\}_{i=1}^{n_B})$ **then**
  return 0;
else
  return 1;

[4] Unfortunately, set *Chck* is undefined in both experiments and set *Reg* is undefined in the latter. Set *Chck* should contain triples of voter identities, their votes, and ballots, for voters that "checked that their ballots will be counted [using mechanisms defined for individual verifiability]," but there is no notion of voters performing individual verifiability checks in either experiment. (Universal verifiability checks are performed.) Set *Reg* is maintained by oracle E, yet "the adversary is not given...access to [this] oracle [in the latter experiment], since it controls the registrar and thus can register users arbitrarily, even with malicious credentials." No alternative definition of set *Reg* is given, hence, the set is undefined for oracles C and R. Since our results are not reliant on set *Chck* nor *Reg*, we will not second-guess the authors' intention, and we leave undefined sets in experiments to signal that something is missing.

$\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}g}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k) =$

$(PK_{\mathcal{T}}, SK_{\mathcal{T}}) \leftarrow \mathsf{Setup}(k);$
$BB \leftarrow \emptyset;\ Crpt \leftarrow \emptyset;\ Rvld \leftarrow \emptyset;$
$(X, P) \leftarrow \mathcal{A}^{B,C,R}(PK_{\mathcal{T}});$
if $\mathsf{Verify}(PK_{\mathcal{T}}, BB, X, P) = 0$ **then return** 0;
if $X = \perp$ **then return** 0;
if $\exists (i_1^A, \beta_1^A, b_1^A), \ldots, (i_{n_A}^A, \beta_{n_A}^A, b_{n_A}^A) \in Rvld \setminus Chck$
$\wedge\ \exists \beta_1^B, \ldots, \beta_{n_B}^B\ .\ 0 \leq n_B \leq |Crpt|$
$\wedge\ X = \rho(\{\beta \mid (i, \beta, b) \in Chck\}) \star \rho(\{\beta_i^A\}_{i=1}^{n_A}) \star \rho(\{\beta_i^B\}_{i=1}^{n_B})$ **then**
  return 0;
else
  return 1;

Oracle E is used to model $\mathcal{A}$ enrolling voters. On invocation $E(i)$, oracle E does the following: Computes $(pk, sk) \leftarrow \mathsf{Register}(PK_{\mathcal{T}}, i, k)$, records i as being enrolled by updating Reg to be $Reg \cup \{(i, pk, sk)\}$, and outputs pk.

Oracle C is used to model $\mathcal{A}$ corrupting voters and learning their private credentials. On invocation $C(\ell)$, if $(\ell, pk, sk) \in Reg$, then the oracle records that voter ℓ is corrupted by updating $Crpt$ to be $Crpt \cup \{(\ell, pk)\}$ and outputs sk.

Oracle R reveals ballots. On invocation $R(i, \beta)$, if $\beta \in \mathbb{V}$ and there exists pk and sk such that $(i, pk, sk) \in Reg \wedge (i, pk) \notin Crpt$, then oracle R does the following: Computes $b \leftarrow \mathsf{Vote}(i, pk, sk, PK_{\mathcal{T}}, \beta)$, records b as being revealed by updating $Rvld$ to be $(Rvld \setminus \{(i, \beta', b') \mid (i, \beta', b') \in Rvld\}) \cup \{(i, \beta, b)\}$, and outputs b.

Oracle B controls the bulletin board. On invocation $B(b)$, oracle B computes $BB \leftarrow \mathsf{Box}(PK_{\mathcal{T}}, BB, b)$.

6.3 Proof of Proposition 3

We define key leakage for Cortier et al. election schemes before proving Proposition 3.

Definition 5. *An election scheme $\mathbb{V}$, $\mathbb{R}$, ρ, $\Pi = (\mathsf{Setup}, \ldots)$* does not leak the tallier's private key *if for security parameters k and probabilistic polynomial-time adversary $\mathcal{A}$, we have $\mathsf{Succ}(\mathsf{Exp\text{-}leak}(\Pi, \mathcal{A}, k))$ is negligible, where:*

$\mathsf{Exp\text{-}leak}(\Pi, \mathcal{A}, k) =$

$(PK_{\mathcal{T}}, SK_{\mathcal{T}}) \leftarrow \mathsf{Setup}(k);$
$SK'_{\mathcal{T}} \leftarrow \mathcal{A}(PK_{\mathcal{T}});$
if $SK'_{\mathcal{T}} = SK_{\mathcal{T}}$ **then**
  return 1;
else
  return 0;

Proof of Proposition 3. We first show that $\mathbb{V}$, $\mathbb{R}$, ρ, $\mathsf{Backdoor}'(\Pi)$ is an election scheme—i.e., it satisfies Cortier et al. correctness and partial tallying. Since Π

does not leak the tallier's private key, it follows that schemes Π and $\mathsf{Backdoor}'(\Pi)$ are equivalent in the context of correctness (in particular, the tallier's private key cannot appear on the bulletin board), hence, $\mathsf{Backdoor}'(\Pi)$ satisfies correctness because Π does. Moreover, since the most significant bit of an honestly generated private key is one, it follows that schemes Π and $\mathsf{Backdoor}'(\Pi)$ are equivalent in the context of partial tallying, hence, $\mathsf{Backdoor}'(\Pi)$ satisfies partial tallying because Π does. Next, we show that Cortier et al. verifiability is satisfied. Since Π does not leak the tallier's private key, experiments $\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \mathsf{Backdoor}'(\Pi), \star, \mathcal{A}, k)$ and $\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}g}(\mathbb{V}, \rho, \mathsf{Backdoor}'(\Pi), \star, \mathcal{A}, k)$ produce bulletin boards that exclude the tallier's private key with overwhelming probability, for all k and probabilistic polynomial-time adversary $\mathcal{A}$. It follows that an execution of experiment $\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \mathsf{Backdoor}'(\Pi), \star, \mathcal{A}, k)$ is an execution of experiment $\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k)$ with non-negligible probability. Similarly, an execution of experiment $\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}g}(\mathbb{V}, \rho, \mathsf{Backdoor}'(\Pi), \star, \mathcal{A}, k)$ is an execution of the experiment $\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}g}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k)$. Thus, $\mathsf{Backdoor}'(\Pi)$ satisfies Cortier et al. verifiability. $\square$

6.4 Proof sketch of Proposition 5

Suppose $\Pi = (\mathsf{Setup}, \dots, \mathsf{Vote}, \dots, \mathsf{Tally}, \mathsf{Verify})$ and $\mathsf{Selective}(\Pi, \varepsilon) = (\mathsf{Setup}, \dots, \mathsf{Vote}, \dots, \mathsf{Tally}, \mathsf{Verify}_R)$. We first show that $\mathbb{V}$, $\mathbb{R}$, ρ, $\mathsf{Selective}(\Pi, \varepsilon)$ is an election scheme—i.e., it satisfies Cortier et al. correctness and partial tallying. Satisfaction of the latter is trivial, since we only modify Verify, and we proceed with a proof of the former: Since $\mathbb{V}$, $\mathbb{R}$, ρ, Π satisfies Cortier et al. correctness and ε is not in Vote's co-domain, we have for all key pairs $(PK_{\mathcal{T}}, SK_{\mathcal{T}})$ output by Setup, subsets BB of Vote's co-domain and tallies (X, P) output by Tally that $\mathsf{Verify}(PK_{\mathcal{T}}, BB, X, P) = 1$ implies $\mathsf{Verify}_R(PK_{\mathcal{T}}, BB, X, P) = 1$. It follows that $\mathbb{V}$, $\mathbb{R}$, ρ, $\mathsf{Selective}(\Pi, \varepsilon)$ satisfies Cortier et al. correctness. Next, we show that Cortier et al. verifiability is satisfied.

Let E hold in an execution of $\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}$ if $\varepsilon \in BB$, where BB is output by the adversary. By definition of $\mathsf{Selective}(\Pi, \varepsilon)$, we have $\Pr[\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \mathsf{Selective}(\Pi, \varepsilon), \star, \mathcal{A}, k) = 1 \mid E] = 0$ and

$$\begin{aligned}\Pr[&\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \mathsf{Selective}(\Pi, \varepsilon), \star, \mathcal{A}, k) = 1 \mid \neg E] \\ &= \Pr[\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k) = 1 \mid \neg E].\end{aligned}$$

It follows that

$$\begin{aligned}&\mathsf{Succ}(\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \mathsf{Selective}(\Pi, \varepsilon), \star, \mathcal{A}, k)) \\ &\qquad = \Pr[\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k) = 1 \mid \neg E].\end{aligned}$$

Since

$$\begin{aligned}\mathsf{Succ}(&\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k)) = \\ &\Pr[\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k) = 1 \mid E] \\ &\quad + \Pr[\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k) = 1 \mid \neg E],\end{aligned}$$

we have

$$\begin{aligned}&\mathsf{Succ}(\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \mathsf{Selective}(\Pi, \varepsilon), \star, \mathcal{A}, k))\\ &\qquad\qquad\leq \mathsf{Succ}(\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}b}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k)).\end{aligned}$$

Similarly, we derive

$$\begin{aligned}&\mathsf{Succ}(\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}g}(\mathbb{V}, \rho, \mathsf{Selective}(\Pi, \varepsilon), \star, \mathcal{A}, k))\\ &\qquad\qquad\leq \mathsf{Succ}(\mathsf{Exp\text{-}CGGI\text{-}Ver\text{-}g}(\mathbb{V}, \rho, \Pi, \star, \mathcal{A}, k)).\end{aligned}$$

Since $\mathbb{V}$, $\mathbb{R}$, ρ, Π satisfies Cortier et al. verifiability, we have $\mathbb{V}$, $\mathbb{R}$, ρ, $\mathsf{Selective}(\Pi, \varepsilon)$ satisfies Cortier et al. verifiability too. □

7 Conclusion

We have seen that definitions of verifiability by Juels et al., Cortier et al. and Kiayias et al. do not detect some collusion, biasing and malicious-key attacks. Although a well-designed voting system would hopefully not exhibit vulnerabilities to these attacks, it is the job of security definitions to detect malicious systems, regardless of whether vulnerabilities are due to malice or error. So definitions of election verifiability should preclude them. Yet, the aforementioned definitions do not. Revisiting the security of previously-analysed voting systems in light of the attacks identified here is a possible direction for future work. Our findings have been reported to original authors Dario Catalano, Véronique Cortier, Markus Jakobsson, and David Galindo, and will fuel the exploration for a new definition of verifiability. In collaboration with Frink, we have developed one such definition.

Acknowledgements We thank Dario Catalano, Jeremy Clark, Véronique Cortier, Aleksander Essex, Steven Frink, David Galindo, Stéphane Glondu, Markus Jakobsson, Steve Kremer, and Mark Ryan for insightful discussions that have influenced this manuscript. We also thank our IPL reviewers and editors, whose commentary led to significant improvements. This work is partly supported by AFOSR grants FA9550-12-1-0334 and FA9550-14-1-0334, by NSF grant 1421373, by the National Security Agency, and the Luxembourg National Research Fund (FNR) under the FNR-INTER-VoteVerif project (10415467).

References

[Adi06] Ben Adida. *Advances in Cryptographic Voting Systems*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2006.

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

[BT94] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC '94: Twenty-sixth Annual ACM Symposium on Theory of Computing*, pages 544–553, New York, NY, USA, 1994. ACM Press.

[CF85] Josh Daniel Cohen and Michael J. Fischer. A Robust and Verifiable Cryptographically Secure Election Scheme. In *FOCS'85: 26th IEEE Symposium on Foundations of Computer Science*, pages 372–382. IEEE Computer Society, 1985.

[CGGI14] Véronique Cortier, David Galindo, Stéphane Glondu, and Malika Izabachène. Election Verifiability for Helios under Weaker Trust Assumptions. In *ESORICS'14: 19th European Symposium on Research in Computer Security*, volume 8713 of *LNCS*, pages 327–344. Springer, 2014. Accompanied by INRIA technical report RR-8555.

[CRS05] David Chaum, Peter Y. A. Ryan, and Steve Schneider. A Practical Voter-Verifiable Election Scheme. In *ESORICS'05: 10th European Symposium On Research In Computer Security*, volume 3679 of *LNCS*, pages 118–139. Springer, 2005.

[Dag07] Participants of the Dagstuhl Conference on Frontiers of E-Voting. *Dagstuhl Accord*, 2007. `http://dagstuhlaccord.org/`.

[JCJ10] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. In *Towards Trustworthy Elections: New Directions in Electronic Voting*, volume 6000 of *LNCS*, pages 37–63. Springer, 2010.

[JS12] Douglas W. Jones and Barbara Simons. *Broken Ballots: Will Your Vote Count?*, volume 204 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, 2012.

[KRS10] Steve Kremer, Mark D. Ryan, and Ben Smyth. Election verifiability in electronic voting protocols. In *ESORICS'10: 15th European Symposium on Research in Computer Security*, volume 6345 of *LNCS*, pages 389–404. Springer, 2010.

[KSRW04] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach. Analysis of an Electronic Voting System. In *S&P'04: 25th Security and Privacy Symposium*, pages 27–40. IEEE Computer Society, 2004.

[KZZ15] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. End-to-end verifiable elections in the standard model. In *EUROCRYPT'15: 34th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 9057 of *LNCS*, pages 468–498. Springer, 2015.

[SFC15] Ben Smyth, Steven Frink, and Michael R. Clarkson. Election Verifiability: Cryptographic Definitions and an Analysis of Helios and JCJ. IACR ePrint 2015/233, 2015.

[Smy20] Ben Smyth. Surveying global verifiability. *Information Processing Letters*, 163, 2020.

[WWH+10] Scott Wolchok, Eric Wustrow, J. Alex Halderman, Hari K. Prasad, Arun Kankipati, Sai Krishna Sakhamuri, Vasavya Yagati, and Rop Gonggrijp. Security Analysis of India's Electronic Voting Machines. In *CCS'10: 17th ACM Conference on Computer and Communications Security*, pages 1–14. ACM Press, 2010.