Surveying definitions of coercion resistance

Ben Smyth

Interdisciplinary Centre for Security, Reliability and Trust,
University of Luxembourg, Esch-sur-Alzette, Luxembourg

July 15, 2019

Abstract

We explore definitions of coercion resistance in the computational model of cryptography; discovering all but one are too weak (i.e., satisfiable by voting systems that are not coercion resistant) and the other is too strong (i.e., unsatisfiable by voting systems that are). Hence, we show that coercion resistance has not been adequately formalised. Our results cast doubt over the security of voting systems that have been proven secure with respect to those definitions; a new definition is necessary.

1 Introduction

Coercion resistance is the strongest notion of privacy a voting system can deliver. It asserts that no voter can prove they followed a coercer's instructions; ensuring voters can evade coercion and vote freely. Coercion resistance was introduced by Okamoto [Oka98] and first formalised by Juels, Catalano & Jakobsson [JCJ02, JCJ05, JCJ10]. It strengthens receipt-freeness by considering an adversary that instructs a voter – possibly with instructions that deviate from the prescribed ballot casting procedure – rather than merely asking the voter for proof [MN06, KZZ15, CCFG16, FQS19]. In turn, receipt-freeness strengthens ballot secrecy, wherein the adversary's communication capabilities are limited to controlling ballot collection [Smy19]. In its strongest form, coercion resistance includes protection against forced abstention attacks, whereby a coercer instructs a voter to abstain, yet the voter is able to evade coercion and vote freely (assuming the coercer does not control all ballot-collection channels).

Following the first definition of coercion resistance by Juels, Catalano & Jakobsson, further definitions have been proposed by Gardner, Garera & Rubin [GGR09], Unruh & Müller-Quade [UM10], and Küsters, Truderung & Vogt [KTV10, KTV12]. We will explore the three most recent definitions, in the context of syntax by Juels, Catalano & Jakobsson, which is common to each. Indeed, the definition by Küsters, Truderung & Vogt is stated independently of any particular syntax and the definition by Gardner, Garera & Rubin is largely syntax independent, hence, those definitions can be considered in the narrower context of syntax by Juels, Catalano & Jakobsson. The definition by Unruh & Müller-Quade is stated in terms of a particular syntax (but to a lesser extent than the definition by Juels, Catalano & Jakobsson) and we cast their definition into the context of syntax by Juels, Catalano & Jakobsson. Using a common syntax simplifies our exploration and facilitates comparisons between definitions.

Contribution and structure. We critic definitions of coercion resistance by Gardner, Garera & Rubin (§3), Unruh & Müller-Quade (§4), and Küsters, Truderung & Vogt (§5), discovering all but the final definition are too weak (i.e., satisfiable by voting systems that are not coercion resistant) and the other – by Küsters, Truderung & Vogt – is too strong (i.e., unsatisfiable by voting systems that are). Hence, we show that coercion resistance has not been adequately formalised. Our results cast doubt over the security of voting systems that have been proven secure with respect

Sidebar 1 Preliminaries: Notation and games [Smy19]

We let $A(x_1, \ldots, x_n; r)$ denote the output of probabilistic algorithm A on inputs $x_1, \ldots, x_n$ and coins r, and we let $A(x_1, \ldots, x_n)$ denote $A(x_1, \ldots, x_n; r)$, where coins r are chosen uniformly at random from the coin space of algorithm A. Moreover, we let $x \leftarrow T$ denote assignment of T to x, and $x \leftarrow_R S$ denote assignment to x of an element chosen uniformly at random from set S, similarly, $x \leftarrow_R D$ denotes assignment to x of an element chosen uniformly at random from distribution D. Furthermore, we let $x[i]$ denote component i of vector x and let $|x|$ denote the length of vector x. Finally, we write $(x_1, \ldots, x_{|T|}) \leftarrow T$ for $x \leftarrow T; x_1 \leftarrow x[1]; \ldots; x_{|T|} \leftarrow x[|T|]$, when T is a vector, and $x, x' \leftarrow_R S$ for $x \leftarrow_R S; x' \leftarrow_R S$.

A game is a probabilistic algorithm that output a boolean. Using our notation, we can formulate the following game, denoted $\mathsf{Exp}(H, S, \mathcal{A})$, which tasks an adversary $\mathcal{A}$ to distinguish between a function H and a simulator S: $m \leftarrow \mathcal{A}(); \beta \leftarrow_R \{0,1\}$; **if** $\beta = 0$ **then** $x \leftarrow H(m)$ **else** $x \leftarrow S(m)$; $g \leftarrow \mathcal{A}(x)$; **return** $g = \beta$. Adversaries are *stateful*, i.e., information persists across invocations of an adversary in a game. In particular, adversaries can access earlier assignments. For instance, the adversary's second instantiation in game Exp has access to any assignments made during its first instantiation. An adversary *wins* a game by causing it to output true ($\top$) and the adversary's *success* in a game $\mathsf{Exp}(\cdot)$, denoted $\mathsf{Succ}(\mathsf{Exp}(\cdot))$, is the probability that the adversary wins, that is, $\mathsf{Succ}(\mathsf{Exp}(\cdot)) = \Pr[\mathsf{Exp}(\cdot) = \top]$. We focus on computational security, rather than information-theoretic security, and tolerate breaks by adversaries in non-polynomial time and breaks with negligible success, since such breaks are infeasible in practice. Game Exp captures a single interaction between the challenger and the adversary. We can extend games with oracles to capture arbitrarily many interactions. For instance, we can formulate a strengthening of Exp as follows: $\beta \leftarrow_R \{0,1\}; g \leftarrow \mathcal{A}^{\mathcal{O}}(x)$; **return** $g = \beta$, where $\mathcal{A}^{\mathcal{O}}$ denotes $\mathcal{A}$'s access to oracle $\mathcal{O}$ and $\mathcal{O}(m)$ computes **if** $\beta = 0$ **then** $x \leftarrow H(m)$ **else** $x \leftarrow S(m)$; **return** x. Oracles may access game parameters such as bit β.

to those definitions; a new definition is necessary. The remaining sections introduce syntax (§2) and present a brief conclusion (§6), and Sidebar 1 introduces games and standard notation.

2 Election scheme syntax

We will consider definitions of coercion resistance in the context of syntax by Juels, Catalano & Jakobsson, more precisely, we adopt the variant of their syntax by Smyth, Frink & Clarkson [SFC18], which clarifies a few details. The syntax captures a class of voting systems that consist of the following four steps. First, a tallier generates a key pair. Secondly, a registrar generates credentials for voters.[1] Thirdly, each voter constructs and casts a ballot for their vote. These ballots are recorded on a bulletin board. Finally, the tallier tallies the recorded ballots and announces the outcome as a frequency distribution of votes. (The chosen representative is derived from this distribution, which suffices for both first-past-the-post and ranked-choice voting systems [Smy18]).)

Definition 1 (Election scheme [SFC18])**.** *An* election scheme *is a tuple of probabilistic polynomial-time algorithms* $(\mathsf{Setup}, \mathsf{Register}, \mathsf{Vote}, \mathsf{Tally})$ *such that:*[2]

Setup*, denoted* $(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa)$*, is run by the tallier. The algorithm takes a security parameter* κ *as input and outputs a key pair* pk, sk*, a maximum number of ballots* mb*, and a maximum number of candidates* mc*.*

[1] Smyth proves that election schemes without registration – i.e., *election schemes with external authentication* [SFC18] – do not satisfy receipt-freeness nor coercion resistance [Smy19, §7].

[2] The syntax bounds the number of ballots mb, respectively candidates mc, to broaden the correctness definition's scope. The syntax represents votes as integers, rather than alphanumeric strings, for brevity. Finally, the syntax employs sets, rather than multisets or lists, to preclude construction (and consequently modelling) of schemes vulnerable to attacks that arise due to duplicate ballots.

Register, *denoted* $(pd, sd) \leftarrow \mathsf{Register}(pk, \kappa)$*, is run by the registrar. The algorithm takes a public key* pk *and security parameter* κ *as input and outputs a public credential* pd *and a private credential* sd*.*

Vote, *denoted* $b \leftarrow \mathsf{Vote}(sd, pk, v, nc, \kappa)$*, is run by voters. The algorithm takes as input a private credential* sd*, a public key* pk*, a voter's vote* v*, some number of candidates* nc*, and a security parameter* κ*. The vote should be selected from a sequence* $1, \ldots, nc$ *of candidates. The algorithm outputs a ballot* b *or error symbol* $\perp$*.*

Tally, *denoted* $(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, L, nc, \kappa)$*, is run by the tallier. The algorithm takes as input a private key* sk*, a bulletin board* $\mathfrak{bb}$*, an electoral roll* L*, some number of candidates* nc*, and a security parameter* κ*, where* $\mathfrak{bb}$ *and* L *are sets. And outputs an election outcome* $\mathfrak{v}$ *and a non-interactive tallying proof* pf *demonstrating that the outcome corresponds to votes expressed in ballots on the bulletin board. The election outcome* $\mathfrak{v}$ *should be a vector of length* nc *such that* $\mathfrak{v}[v]$ *indicates the number of votes for candidate* v*.*

Election schemes must satisfy correctness*: there exists a negligible function* negl*, such that for all security parameters* κ*, integers* nb *and* nc*, and votes* $v_1, \ldots, v_{nb} \in \{1, \ldots, nc\}$*, it holds that, given a zero-filled vector* $\mathfrak{v}$ *of length* nc*, we have:* $\Pr[(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa);$ **for** $1 \leq i \leq nb$ **do** $\{ (pd_i, sd_i) \leftarrow \mathsf{Register}(pk, \kappa); b_i \leftarrow \mathsf{Vote}(sd_i, pk, v_i, nc, \kappa); \mathfrak{v}[v_i] \leftarrow \mathfrak{v}[v_i] + 1; \}$ $(\mathfrak{v}', pf) \leftarrow \mathsf{Tally}(sk, \{b_1, \ldots, b_{nb}\}, \{pd_1, \ldots, pd_{nb}\}, nc, \kappa) : nb \leq mb \wedge nc \leq mc \Rightarrow \mathfrak{v} = \mathfrak{v}'] > 1 - \mathsf{negl}(\kappa)$.

Some voting systems (e.g., the system by Juels, Catalano & Jakobsson) permit the tallier's role to be distributed amongst several talliers. This improves security, because a single rogue tallier cannot tally individual ballots to reveal votes. But, it also increases complexity. We consider a single tallier to avoid such complexities. Moreover, we omit algorithm Verify from our syntax, because we focus on privacy, rather than verifiability, in this paper.

Definitions of coercion resistance are reliant on candidate ϕ representing abstention, so we include ϕ in the sequence $1, \ldots, nc$ of candidates, in the context of algorithm Vote.

3 Gardner, Garera & Rubin [GGR09]

Gardner, Garera & Rubin formulate a game-based definition of coercion resistance that challenges an adversary to distinguish a ballot for the adversary's preferred vote v_0 from a ballot for the voter's preferred vote v_1, wherein ballots are constructed using coins provided by the adversary and the latter ballot is constructed using inputs that may have been modified for the purposes of evading coercion.

Definition 2. *Let* $\Gamma = (\mathsf{Setup}, \mathsf{Register}, \mathsf{Vote}, \mathsf{Tally})$ *be an election scheme,* $\mathcal{A}$ *be an adversary, and* κ *be a security parameter. Furthermore, let* generateInput *be an algorithm that takes the inputs to algorithm* Vote *and some coins as input, and outputs a private credential, a public key, an integer, and some coins. We say* Γ *satisfies* GGR-CR*, if there exists a probabilistic polynomial-time algorithm* generateInput *such that for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, there exists a negligible function* negl *and for all security parameters* κ*, we have* $\mathsf{Succ}(\mathsf{GGR\text{-}CR}(\Gamma, \mathcal{A}, \mathsf{generateInput}, \kappa)) < \frac{1}{2} + \mathsf{negl}(\kappa)$*, where game* GGR-CR *is defined in Figure 1.*

Game GGR-CR proceeds as follows: The challenger generates a key pair and a credential (Lines 1 & 2), and initialises an empty set of coins (Line 3). Next, the adversary chooses their preferred vote, the voter's preferred vote, some number of candidates, and some coins (Line 4).[3,4]

[3] Gardner, Garera & Rubin do not specify the adversary's precise inputs used (by the challenger and oracle) to construct ballots. We presume that the coerced voter controls some of the inputs to algorithm Vote, in particular, we presume that voter controls their private credential and public information, including the public key and the security parameter.

[4] Gardner, Garera & Rubin consider voting systems in which ballots can contain some information that is only available to the verifier, whereas such ballots are excluded by our election scheme syntax. Hence, we omit the verifier from our formalisation.

Figure 1 Game GGR-CR

$\mathsf{GGR\text{-}CR}(\Gamma, \mathcal{A}, \mathsf{generateInput}, \kappa) =$

1 $(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa)$;
2 $(pd, sd) \leftarrow \mathsf{Register}(pk, \kappa)$;
3 $\mathfrak{r} \leftarrow \emptyset$;
4 $(v_0, v_1, nc, s) \leftarrow \mathcal{A}^{\mathcal{O}}(pk, pd)$;
5 $\mathfrak{r} \leftarrow \mathfrak{r} \cup \{s\}$;
6 $\beta \leftarrow_R \{0, 1\}$;
7 **if** $\beta = 0$ **then**
8 $\quad (sd', pk', nc', \kappa') \leftarrow \mathsf{generateInput}(sd, pk, v_0, nc, \kappa, s, v_1)$;
9 $\quad b \leftarrow \mathsf{Vote}(sd', pk', v_1, nc', \kappa'; s)$;
10 **else**
11 $\quad b \leftarrow \mathsf{Vote}(sd, pk, v_0, nc, \kappa; s)$;
12 $g \leftarrow \mathcal{A}^{\mathcal{O}}(b)$;
13 **return** $g = \beta \wedge s \notin \mathfrak{r}$;

Oracle $\mathcal{O}$ is defined such that $\mathcal{O}(v, nc, r)$ computes $b \leftarrow \mathsf{Vote}(sd, pk, v, nc, \kappa; r); \mathfrak{r} \leftarrow \mathfrak{r} \cup \{r\}$ and outputs b.

The challenger adds those coins to the set of coins (Line 5).[5] The challenger flips a coin (Line 6), constructs a ballot on behalf of the coerced voter using inputs that may have been modified if the coin flip produces zero (Lines 7–9) and the adversary's input otherwise (Lines 10 & 11). Finally, the adversary is given the constructed ballot and attempts to determine whether the coin flip resulted in zero or one (Lines 12 & 13).

The definition is too weak. In particular, tallying may leak information that can violate privacy. For instance, as an extreme example, suppose tallying leaks the tallier's private key, thereby enabling tallying of individual ballots to reveal each voter's vote. Perhaps Gardner, Garera & Rubin intended to capture coercion resistance of ballot casting, rather than coercion resistance of the entire voting system. Indeed, they write, "We introduce a new definition of a coercion resistant *vote casting* protocol" (emphasis added). Yet, they go on to write, "we are able to address coercion enabled by examination of the protocol's final output," which suggests that the final output – surely including the election outcome and tallying proof – should have been considered. Moreover, they are critical of the definition by Moran & Naor [MN06] because "[it] focuses on the adversary's view of a voter's interactions with a machine and allows privacy leaks in the final output in the protocol," which seemingly suggests their definition should detect such leaks.

We shared our findings with Garera & Rubin (email, 6 Jul 2018), but have not received a response.

4 Unruh & Müller-Quade [UM10]

Unruh & Müller-Quade formulate a game-based definition of coercion resistance that challenges an adversary to distinguish between a voter following a coercer's instructions to cast a vote v^* preferred by the adversary and the voter deviating from those instructions to cast a vote v preferred by the voter (whilst producing evidence that the instructions were followed), with probability greater than the adversary's ability to distinguish the election outcomes produced in each setting. The definition requires a counter-strategy that deviates from the adversary's instructions, which is captured using an algorithm C. That counter-strategy must produce a ballot for v.

[5] The presentation by Gardner, Garera & Rubin does not seem to include the adversary's coins in the set of coins, yet this is necessary to ensure the definition is satisfiable. We believe this is an omission and we include the coins in the set.

Figure 2 Games UM-CR and UM-CR-$, with distinctions highlighted in yellow

$\mathsf{UM\text{-}CR}(\Gamma, \mathcal{A}, nc, nv, v, v^*, D, j, \kappa) =$

1 $(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa)$;
2 **for** $1 \leq i \leq nv$ **do**
3 $\quad (pd_i, sd_i) \leftarrow \mathsf{Register}(pk, \kappa)$;
4 $L \leftarrow \{pd_1, \ldots, pd_{nv}\}$;
5 $\mathfrak{bb} \leftarrow \emptyset$;
6 **if** $v^* \neq \phi$ **then**
7 $\quad b \leftarrow \mathsf{Vote}(sd_j, pk, v^*, nc, \kappa)$;
8 $\quad \mathfrak{bb} \leftarrow \mathfrak{bb} \cup \{b\}$;
9 $x \leftarrow \mathcal{A}^{\mathcal{O}}(pk, v^*, L, \kappa)$;
10 $\mathfrak{v} \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, L, nc, \kappa)$;
11 $g \leftarrow \mathcal{A}(\mathfrak{v})$;
12 **return** g;

$\mathsf{UM\text{-}CR\text{-}\$}(\Gamma, C, \mathcal{A}, nc, nv, v, v^*, D, j, \kappa) =$

1 $(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa)$;
2 **for** $1 \leq i \leq nv$ **do**
3 $\quad (pd_i, sd_i) \leftarrow \mathsf{Register}(pk, \kappa)$;
4 $L \leftarrow \{pd_1, \ldots, pd_{nv}\}$;
5 $\mathfrak{bb} \leftarrow \emptyset$;
6 **if** $v^* \neq \phi$ **then**
7 $\quad b \leftarrow C(sd_j, pk, v, v^*, nc, \kappa)$;
8 $\quad \mathfrak{bb} \leftarrow \mathfrak{bb} \cup \{b\}$;
9 $x \leftarrow \mathcal{A}^{\mathcal{O}}(pk, v^*, L, \kappa)$;
10 $\mathfrak{v} \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, L, nc, \kappa)$;
11 $g \leftarrow \mathcal{A}(\mathfrak{v})$;
12 **return** g;

Oracle $\mathcal{O}$ is defined such that $\mathcal{O}(i)$ computes $v \leftarrow_R D$; **if** $v \neq \phi$ **then** $b \leftarrow \mathsf{Vote}(sd_i, pk, v, nc, \kappa)$; $\mathfrak{bb} \leftarrow \mathfrak{bb} \cup \{b\}$, where $i \in \{1, \ldots, nv\} \setminus \{j\}$. Moreover, we require that $1 \leq j \leq nv$ and that oracle $\mathcal{O}$ is called with integer i at most once.

Definition 3. *Let* $\Gamma = (\mathsf{Setup}, \mathsf{Register}, \mathsf{Vote}, \mathsf{Tally})$ *be an election scheme,*[6] $\mathcal{A}$ *be an adversary, and* κ *be a security parameter. Moreover, let* nc *and* nv *be integers,* $v, v^* \in \{1, \ldots, nc\}$ *be votes,* $j \in \{1, \ldots, nv\}$ *be a voter's identity, and* D *be an distribution over* $\{1, \ldots, nc, \phi\}^{nv-1}$. *We define games* UM-CR *and* UM-CR-$ *in Figure 2, and say* Γ *satisfies* UM-CR *with respect to* nc, nv*, if there exists a probabilistic polynomial-time algorithm* C *such that for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, efficiently sampleable distributions* D *over* $\{1, \ldots, nc, \phi\}^{nv-1}$*, votes* $v, v^* \in \{1, \ldots, nc, \phi\}$*, and integers* $j \in \{1, \ldots, nv\}$*, there exists a negligible function* negl *and the following conditions hold for all security parameters* κ*: First, the election outcome* $\mathfrak{v}$ *computed in game* $\mathsf{UM\text{-}CR}(\Gamma, \mathcal{A}, nc, nv, v, v^*, D, j, \kappa)$ *is computationally indistinguishable from the election outcome computed from the votes sampled by the oracle in game* $\mathsf{UM\text{-}CR\text{-}\$}(\Gamma, C, \mathcal{A}, nc, nv, v, v^*, D, j, \kappa)$ *and the vote* v*. Secondly,* $|\mathsf{UM\text{-}CR}(\Gamma, \mathcal{A}, nc, nv, v, v^*, D, j, \kappa) - \mathsf{UM\text{-}CR\text{-}\$}(\Gamma, C, \mathcal{A}, nc, nv, v, v^*, D, j, \kappa)| \leq \max_{v^* \in \{1, \ldots, nc, \phi\}} \Delta(D_v, D_{v^*}) + \mathsf{negl}(\kappa)$*, where* Δ *denotes statistical distance and* D_v*, respectively* D_{v^*}*, is the distribution over* $\{1, \ldots, nc, \phi\}^{nv}$ *that chooses* $nv - 1$ *votes according to* D *and uses* v*, respectively* v^**, for the* nvth *vote.*

The first condition captures the counter-strategy, represented as algorithm C, casting a ballot for v, and the second condition captures the adversary's inability to distinguish the jth voter following instructions (UM-CR) and deviating from them using algorithm C (UM-CR-$).

The games are identical except for Line 7: The challenger generates a key pair and credentials (Lines 1–4),[7] and initialises an empty bulletin board (Line 5). Moreover, if adversarial preferred vote v^* does not represent abstention, then the challenger constructs a ballot for that vote in game UM-CR and constructs a ballot using the counter-strategy in game UM-CR-$,[8] the constructed ballot is added to the bulletin board in both games (Lines 6–8). Next, the adversary instructs the oracle to construct ballots on behalf of non-coerced voters and add those ballots to the bulletin board (Line 9).[9] Finally, the challenger tallies the bulletin board and the adversary is given the election outcome and challenged to determine whether the ballot constructed by the adversary

[6] Unruh & Müller-Quade consider a tallying algorithm that only outputs an election outcome, hence, we consider a variant of algorithm Tally, denoted $\mathfrak{v} \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, L, nc, \kappa)$, in this section.

[7] Key generation is implicit, rather than explicit, in the original presentation by Unruh & Müller-Quade.

[8] Unruh & Müller-Quade suggest that the adversary *instructs* the coerced voter, but do not specify further details. Here, we assume the adversary's ability to instruct is limited to choosing the adversarial preferred vote, which we model by universal quantification over all votes.

[9] The oracle is not explicitly defined by Unruh & Müller-Quade, but it is useful to capture their requirements that the adversary controls the start of tallying and that the voters other than the coerced voter all following the protocol.

was for v or v^* (Lines 10–12).

Definition UM-CR is similar to the receipt-freeness definition by Delaune, Kremer & Ryan [DKR06, DKR09], which Smyth casts from the symbolic model to the computational model of cryptography, resulting in a pair of games Receipt-Freeness-A and Receipt-Freeness-B [Smy19, §7].[10] The former captures the adversary's inability to distinguish between voters following instructions and deviating from them using a counter-strategy (which is similar to the second condition of definition UM-CR), and the latter over-approximates the counter-strategy casting a ballot for vote preferred candidates (which is similar to the first condition of definition UM-CR). Hence, definition UM-CR seemingly captures receipt-freeness rather than coercion resistance. But, upon closer inspection, aspects of Smyth's definition are omitted from the definition by Unruh & Müller-Quade. Indeed, Smyth permits the adversary to control ballot collection and to learn coins used to construct ballots, whereas Unruh & Müller-Quade do not. The former is necessary to detect attacks against ballot secrecy (in the presence of an adversary that controls ballot collection) and the latter to detect attacks against receipt-freeness. Consequently, definition UM-CR does not capture ballot secrecy nor receipt-freeness, thus, coercion resistance is not captured either.

We shared our findings with Unruh (email, 29 Jun 2019), but have not received a response.

5 Küsters, Truderung & Vogt [KTV10, KTV12]

Küsters, Truderung & Vogt formulate a game-based definition of coercion resistance that challenges an adversary to distinguish between a voter following a coercer's instructions to give-up their private credential and the voter deviating from those instructions to cast their preferred vote using a counter-strategy whilst giving-up a fake credential.

Definition 4. *Let* $\Gamma = (\mathsf{Setup}, \mathsf{Register}, \mathsf{Vote}, \mathsf{Tally})$ *be an election scheme,* $\mathcal{A}$ *be an adversary,* C *be a (stateful) algorithm, and* κ *be a security parameter. Moreover, let* na*,* nc *and* nv *be integers, and let* D *be an distribution over* $\{1, \ldots, nc, \phi\}$*. We say* Γ *satisfies* δ-KTV-CR *with respect to* na, nc, nv, C, D*, if for all probabilistic polynomial-time algorithms* $\mathcal{A}$ *and votes* $v \in \{1, \ldots, nc, \phi\}$*, there exists a negligible function* negl *such that for all security parameters* κ*, we have* $\mathsf{Succ}(\mathsf{KTV\text{-}CR}(\Gamma, \mathcal{A}, na, nc, nv, C, D, \kappa)) - \mathsf{Succ}(\mathsf{KTV\text{-}CR\text{-}\$}(\Gamma, \mathcal{A}, na, nc, nv, v, C, D, \kappa)) \leq \delta + \mathsf{negl}(\kappa)$*, where games* KTV-CR *and* KTV-CR-$ *are defined in Figure 3.*

Beyond the requirements specified in Definition 4, Küsters, Truderung & Vogt remark that "[algorithm C must be] defined in such a way that...the coerced voter achieves [their] own goal, e.g., votes for [their] favorite candidate, despite what the coercer tells [them] to do. The concrete definition of [C] depends on the specific goals one wants the coerced voter to be able to achieve... We therefore do not fix this...up front." Unfortunately, this makes the definition incomplete and we will see that this can cause problems.

The first four lines games KTV-CR and KTV-CR-$ are identical: The challenger generates a key pair and credentials (Lines 1–3), and initialises an empty bulletin board (Line 4). In game KTV-CR-$, the challenger constructs a ballot using the counter-strategy and adds that ballot to the bulletin board, except if the coerced voter wants to abstain (Lines 5–7). The next seven lines are identical in both games: for each non-corrupt, non-coerced voter (Line 8), a vote is sampled (Line 9), and the challenger constructs and casts a ballot for that vote, except when the vote signifies abstention (Lines 10–12). Moreover, the challenger initialises a set of public credentials (Line 13) and a set of corrupt voters' private credentials (Line 14). Next, the adversary constructs a set of ballots (Lines 15–16), which might include ballots constructed using corrupt voters' private credentials or the coerced voter's private credential in KTV-CR, respectively a fake credential in KTV-CR-$. Finally, the challenger tallies the ballots and the adversary is given the election outcome, along with a proof of correct computation, and is challenged to determine whether the coerced voter gave-up their private credential or followed a strategy to evade coercion (Lines 17–19).

[10] Smyth considers election schemes without registration; it is straightforward to adapt his games to include registration.

Figure 3 Games KTV-CR and KTV-CR-$

KTV-CR$(\Gamma, \mathcal{A}, na, nc, nv, C, D, \kappa) =$

1 $(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa)$;
2 **for** $1 \leq i \leq nv$ **do**
3 $\quad (pd_i, sd_i) \leftarrow \mathsf{Register}(pk, \kappa)$;
4 $\mathfrak{bb} \leftarrow \emptyset$;
5
6
7
8 **for** $na < i < nv$ **do**
9 $\quad v \leftarrow_R D$;
10 $\quad$ **if** $v \neq \phi$ **then**
11 $\quad\quad b \leftarrow \mathsf{Vote}(sd_i, pk, v, nc, \kappa)$;
12 $\quad\quad \mathfrak{bb} \leftarrow \mathfrak{bb} \cup \{b\}$;
13 $L \leftarrow \{pd_1, \ldots, pd_{nv}\}$;
14 $M \leftarrow (sd_1, \ldots, sd_{na})$;
15
16 $\mathfrak{bb} \leftarrow \mathcal{A}(pk, L, M, sd_{nv}, \mathfrak{bb}, \kappa)$;
17 $(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, L, nc, \kappa)$;
18 $g \leftarrow \mathcal{A}(\mathfrak{v}, pf)$;
19 **return** g;

KTV-CR-$$(\Gamma, \mathcal{A}, na, nc, nv, v, C, D, \kappa) =$$

1 $(pk, sk, mb, mc) \leftarrow \mathsf{Setup}(\kappa)$;
2 **for** $1 \leq i \leq nv$ **do**
3 $\quad (pd_i, sd_i) \leftarrow \mathsf{Register}(pk, \kappa)$;
4 $\mathfrak{bb} \leftarrow \emptyset$;
5 **if** $v \neq \phi$ **then**
6 $\quad b \leftarrow C(sd_{nv}, pk, v, nc, \kappa)$;
7 $\quad \mathfrak{bb} \leftarrow \mathfrak{bb} \cup \{b\}$;
8 **for** $na < i < nv$ **do**
9 $\quad v \leftarrow_R D$;
10 $\quad$ **if** $v \neq \phi$ **then**
11 $\quad\quad b \leftarrow \mathsf{Vote}(sd_i, pk, v, nc, \kappa)$;
12 $\quad\quad \mathfrak{bb} \leftarrow \mathfrak{bb} \cup \{b\}$;
13 $L \leftarrow \{pd_1, \ldots, pd_{nv}\}$;
14 $M \leftarrow (sd_1, \ldots, sd_{na})$;
15 $sd \leftarrow C(pk, pd_{nv}, sd_{nv})$;
16 $\mathfrak{bb} \leftarrow \mathcal{A}(pk, L, M, sd, \mathfrak{bb}, \kappa)$;
17 $(\mathfrak{v}, pf) \leftarrow \mathsf{Tally}(sk, \mathfrak{bb}, L, nc, \kappa)$;
18 $g \leftarrow \mathcal{A}(\mathfrak{v}, pf)$;
19 **return** g;

Coercion resistance can be achieved when just two voters are honest and only one is coerced. Indeed, suppose a coerced voter is instructed to abstain and yet they vote, under cover of an honest voter who abstains. If the coercer is unable to distinguish this scenario from the coerced voter following instruction (i.e., abstaining) and the non-coerced, honest voter voting for the coerced voter's preferred vote, then coercion resistance is intuitively achieved. Yet, such scenarios are not captured by δ-KTV-CR; the definition is too strong. Formally, we have the following:

Theorem 1. *No election scheme Γ satisfies δ-KTV-CR with respect to $nv-2, nc, nv, C, D$, where nc and nv are integers, C is an algorithm, and D is a distribution over $\{1, \ldots, nc, \phi\}$ such that for all computations $v \leftarrow_R D$ we have $v = \phi$.*

Proof. Let us consider executions of games KTV-CR and KTV-CR-$ with respect to adversary $\mathcal{A}$ defined such that $\mathcal{A}(pk, L, M, sd, \mathfrak{bb}, \kappa)$ outputs $\emptyset$ and $\mathcal{A}(\mathfrak{v}, pf)$ outputs $|\mathfrak{bb}| + 1 \bmod 2$. Suppose (pk, sk, mb, mc) is an output of $\mathsf{Setup}(\kappa)$, (pd_{nv}, sd_{nv}) is an output of $\mathsf{Register}(pk, \kappa)$, and b is an output of $C(sd_{nv}, pk, v, nc, \kappa)$, where $v \in \{1, \ldots, nc\}$ is a vote and κ is a security parameter. The for-loop on Lines 8–12 of each game executes exactly once, since $i = nv-1$ is the only integer such that $nv-2 < nv-1 < nv$, but the inner if-branch is never executed, because our hypothesis asserts that computations $v \leftarrow_R D$ are such that $v = \phi$. It follows that $\mathsf{Succ}(\text{KTV-CR}(\Gamma, \mathcal{A}, nv-2, nc, nv, C, D, \kappa)) - \mathsf{Succ}(\text{KTV-CR}(\Gamma, \mathcal{A}, nv-2, nc, nv, v, C, D, \kappa)) = |\emptyset| + 1 \bmod 2 - |\{b\}| + 1 \bmod 2 = 1$, hence, δ-KTV-CR is not satisfied. □

This result renders δ-KTV-CR unsuitable, since election schemes should be secure for any election with at least two candidates and at least two voters. We shared our findings with Küsters, Truderung & Vogt (email, 29 Jun 2019), Küsters raised their Bingo Voting, Scantegrity and ThreeBallot case studies (email, 1 Jul 2019), and our response clarified that those studies consider their weaker privacy definition, rather than coercion resistance (email, 1 Jul 2019). We have not received any further communication from Küsters, Truderung, nor Vogt.

Küsters, Truderung & Vogt do not formally restrict the class of counter-strategies that may be encoded into algorithm C. This is dangerous, since inappropriate strategies might be insufficient to ensure coercion resistance. For instance, we might consider algorithm C such that

$C(pk, pd_{nv}, sd_{nv})$ outputs sd_{nv}, and $C(sd_{nv}, pk, v, nc, \kappa)$ outputs a ballot for vote v, i.e., the algorithm computes $b \leftarrow \mathsf{Vote}(sd_{nv}, pk, v, nc, \kappa)$ and outputs b. With this seemingly reasonable counter-strategy, games KTV-CR and KTV-CR-$ can be shown equivalent: Algorithm C ensures that Line 16 of each game are identical, and the election outcomes computed in Line 17 of each game will be identical for suitable distributions D. Thus, games KTV-CR and KTV-CR-$ are equivalent. Yet, coercion resistance is not necessarily achieved. Indeed, an election scheme that counts the last vote cast with each private credential is not coercion resistant using the counter-strategy, because the strategy reveals credentials, which can be used to cast votes, enabling deviations from the coerced voter's own goal. Some possible restrictions on the class of counter-strategies are discussed by Küsters, Truderung & Vogt, e.g., "when following [the] counter-strategy, [a] coerced voter always successfully votes for the candidate of [their] choice, where 'successfully' means that the coerced voter's vote is in fact counted." We extend Definition 4 with this requirement:

Definition 5. *We say an election scheme satisfies* δ-KTV-CR+ *with respect to* na, nc, nv, C, D*, if* δ-KTV-CR *is satisfied (with respect to* na, nc, nv, C, D*) and the election outcome* $\mathfrak{v}$ *computed in game* KTV-CR-$ *is a frequency distribution that includes the coerced voter's preferred vote and the votes sampled during the for-loop.*

This definition (along with Definition 4) leaves analysts with a conundrum: For which parameters should an election scheme be proven secure? A cautious analyst will of course consider all parameters, but they are doomed to fail, since no scheme will satisfy the definition for all parameters. Indeed, games $\mathsf{KTV\text{-}CR}(\Gamma, \mathcal{A}, na, nc, nv, C, D, \kappa)$ and $\mathsf{KTV\text{-}CR\text{-}\$}(\Gamma, \mathcal{A}, na, nc, nv, v, C, D, \kappa)$ are distinguishable when vote v cannot be sampled from distribution D.

6 Conclusion

This work was initiated by a desire to establish formal relations between definitions of coercion resistance, which would have been useful to establish suitability and relative strength. As work progressed, we discovered that all but the definition by Küsters, Truderung & Vogt are satisfiable by voting systems that are not coercion resistant: Our initial desire serves no purpose; formal relations between unsuitable definitions are worthless. Yet, that discovery is more interesting and will bring an end to the use of unsuitable definitions. Unfortunately, it casts doubt over the security of all voting systems proven secure with respect to those definitions, and establishing their security is a possible direction for future research. We also discovered that the definition by Küsters, Truderung & Vogt is too strong. It follows that coercion resistance has not been adequately formalised, which will fuel the exploration for a suitable definition.

Acknowledgements. This work received financial support from the Luxembourg National Research Fund (FNR) under the FNR-INTER-VoteVerif project (10415467).

References

[CCFG16] Pyrros Chaidos, Véronique Cortier, Georg Fuschbauer, and David Galindo. BeleniosRF: A Non-interactive Receipt-Free Electronic Voting Scheme. In *CCS'16: 23rd ACM Conference on Computer and Communications Security*, pages 1614–1625. ACM Press, 2016.

[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.

[FQS19] Ashley Fraser, Elizabeth A. Quaglia, and Ben Smyth. A critique of game-based definitions of receipt-freeness for voting. In *ProveSec'19: 13th International Conference on Provable anc Practical Security*, LNCS. Springer, 2019.

[GGR09] Ryan W. Gardner, Sujata Garera, and Aviel D. Rubin. Coercion Resistant End-to-end Voting. In *FC'09: 13th International Conference on Financial Cryptography and Data Security*, volume 5628 of *LNCS*, pages 344–361. Springer, 2009.

[JCJ02] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. Cryptology ePrint Archive, Report 2002/165, 2002.

[JCJ05] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. In *WPES'05: 4th Workshop on Privacy in the Electronic Society*, pages 61–70. ACM Press, 2005.

[JCJ10] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. In David Chaum, Markus Jakobsson, Ronald L. Rivest, and Peter Y. A. Ryan, editors, *Towards Trustworthy Elections: New Directions in Electronic Voting*, volume 6000 of *LNCS*, pages 37–63. Springer, 2010.

[KTV10] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. A Game-Based Definition of Coercion-Resistance and its Applications. In *CSF'10: 23rd IEEE Computer Security Foundations Symposium*, pages 122–136. IEEE Computer Society, 2010.

[KTV12] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. A Game-Based Definition of Coercion-Resistance and its Applications. *Journal of Computer Security*, 20(6):709–764, 2012.

[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.

[MN06] Tal Moran and Moni Naor. Receipt-Free Universally-Verifiable Voting with Everlasting Privacy. In *CRYPTO'06: 26th International Cryptology Conference*, volume 4117 of *LNCS*, pages 373–392. Springer, 2006.

[Oka98] Tatsuaki Okamoto. Receipt-Free Electronic Voting Schemes for Large Scale Elections. In *SP'97: 5th International Workshop on Security Protocols*, volume 1361 of *LNCS*, pages 25–35. Springer, 1998.

[SFC18] Ben Smyth, Steven Frink, and Michael R. Clarkson. Election Verifiability: Cryptographic Definitions and an Analysis of Helios and JCJ. Cryptology ePrint Archive, Report 2015/233, 2018.

[Smy18] Ben Smyth. First-past-the-post suffices for ranked voting. `https://bensmyth.com/publications/2017-FPTP-suffices-for-ranked-voting/`, 2018.

[Smy19] Ben Smyth. Ballot secrecy: Security definition, sufficient conditions, and analysis of Helios. Cryptology ePrint Archive, Report 2015/942, 2019.

[UM10] Dominique Unruh and Jörn Müller-Quade. Universally Composable Incoercibility. In *CRYPTO'10: 30th International Cryptology Conference*, volume 6223 of *LNCS*, pages 411–428. Springer, 2010.