A Formal Framework for Modelling Coercion Resistance and Receipt Freeness

James Heather
Department of Computing
University of Surrey, UK
Email: j.heather@surrey.ac.uk

Steve Schneider
Department of Computing
University of Surrey, UK
Email: s.schneider@surrey.ac.uk

***Abstract*—Coercion resistance and receipt freeness are critical properties for any voting system. However, many definitions of these properties have been proposed, with varying levels of formality, and there has been little attempt to tie these definitions together or identify relations between them.**

We give here a general framework for specifying different coercion resistance and receipt freeness properties using the process algebra CSP. The framework is general enough to accommodate a wide range of definitions, including dealing with randomization attacks and forced abstention.

We provide models of some simple voting systems, and show how the framework can be used to analyze these models under different definitions of coercion resistance and receipt freeness. Our formalisation highlights the variation between the definitions in the literature.

I. Introduction

Much work has appeared in the academic literature over the last couple of decades concerning secure voting protocols. Many proposed systems have contained assertions that they meet appropriate security guarantees; but the properties that they are supposed to satisfy are often poorly defined, and for the most part any proofs offered have been informal at best.

More recently, there have been attempts to formalize some of the desirable properties of voting systems [1], [2]. These results have been useful, because they have been able to give precise answers to previously vague questions about the security of various systems. The approach has been to construct a model, formalize the relevant property within the model, and prove that the model satisfies the property.

However, since the informal definitions of these properties vary considerably from one paper to another, these formal definitions inevitably capture what is meant by some authors' use of the terms, and not others'; consequently, one can debate whether the formalisms really have captured the 'right' understanding of the various properties.

Our approach here is a little different. We take two of the commonly discussed properties—coercion resistance and receipt freeness—and we construct a framework that is rich enough to cope with a large variety of definitions. This has the advantage of allowing one to formalize any of a large number of definitions of these properties and analyze a voting system to see which of the definitions it satisfies and which it does not.

This paper begins with an overview of several definitions of receipt freeness and coercion resistance found in the literature. We then consider how to model voting systems formally in CSP, and present a formal CSP framework for describing coercion resistance and receipt freeness properties. We apply the framework to characterise several of the properties from the literature. A simplified CSP model of Prêt à Voter is then considered against a range of coercion resistance properties expressed in our framework. Two further examples of voting systems are presented to highlight differences between definitions in the literature. Finally we discuss receipt freeness properties, which are treated as particular cases of coercion resistance.

A. Definitions

Definitions of coercion resistance and receipt freeness are plentiful in the literature, but the definitions are often informal and ambiguous, and rarely do two definitions coincide. In addition, the difference between coercion resistance and receipt freeness is often unclear.

The following definitions are taken from the literature. They are a mixture of definitions of coercion resistance and receipt freeness; once we have seen the flavour of some of these definitions, then we will consider the difference between the two notions.

Definition 1 (Okamoto [RF]): For any two candidates c and c', a voter can vote for c in a way that is consistent (from the coercer's point of view) with having voted for c' [3].

Definition 2 (Delaune/Kremer/Ryan [CR]): Coercion resistance holds if a coerced voter behaving as instructed is indistinguishable from one voting a different way, to a coercer interacting with the voter [2]. (A weaker notion of receipt freeness is also provided, where two ways of voting are indistinguishable without an explicit coercer.) This approach is extended in [4] to handle forced abstention attacks explicitly. The epistemic approach of [5] takes a similar view of coercion-resistance.

Definition 3 (Universal Composability [CR]): Coercion resistance has also been characterised in the Universal Composability framework, for example in [1], [6], [7], [8]. In this approach, a coercion resistant voting scheme includes a definition of the coercion resistance strategy for voters to follow if coerced, and the system is coercion-resistant if an adversary cannot enable a distinguisher to tell the difference between the real system and an idealised system in which voters choose arbitrarily whether or not to obey the coercer. Though in a different setting, this gives the same sense of coercion-resistance as Definition 2.

Definition 4 (Benaloh/Tuinstra [RF]): A voter should be unable to prove that a vote was cast in a particular way [9].

This definition can have at least two interpretations, but we take 'in a particular way' to refer to the value of the vote. In Prêt à Voter [10], for instance, a receipt with a tick in the top box is not evidence of voting in any particular way.

This still leaves the following two interpretations as possibilities:

1) The voter should be unable to show that she voted for a particular candidate c.
2) The voter should be unable to show that she did *not* vote for a particular candidate c.

Note that the second is stronger than the first, and seems to be the strongest interpretation. If a system meets the second, then it must meet the first, for if a voter can show that she voted for c, then she has thereby shown that she did not vote for $c' \neq c$.

For the purposes of this paper, we will use the second interpretation.

Definition 5 (Chaum et al. [RF]): A voting system is receipt free if the receipt leaks no useful information [11].

The definition leaves a fair amount of room for manoeuvre: what qualifies as 'useful information'? At one end of the scale, a receipt that gives the full vote in plaintext is clearly ruled out; at the other end, something like a Prêt à Voter receipt or a ThreeBallot receipt [12], each of which is consistent with a vote for any candidate, presumably does not give useful information.

It seems reasonable to take this as requiring the receipt not to provide the voter with a proof that she did not vote for a particular candidate. The Chaum definition, under this interpretation, is equivalent to the second interpretation of Definition 4.

Definition 6 (Juels/Catalano/Jakobsson [CR]): A voting system is coercion resistant if it "is one in which the user can deceive the adversary into thinking that she has behaved as instructed, when the voter has in fact cast a ballot according to her own intentions." [13]

The issue here is what can qualify as instruction. Since the main thrust of this paper is to discuss different notions of instruction and the consequent strength of the coercion resistance property, this definition could be taken to encompass many of the properties discussed in the paper.

The difference between coercion resistance and receipt freeness is usually phrased in terms of the coercer's ability to interact with the voter during the voting process: coercion resistance includes protection against a coercer who can interact in this way, whereas receipt freeness does not. This is a slippery distinction, for two reasons:

1) Interacting with the voter *before* the voting process, and interacting *during* the voting process, are hard to distinguish cleanly. For instance, there is nothing in principle to stop the coercer from interacting before voting takes place, and providing the voter with a flowchart showing how the voter is to act in any given situation.
2) It is not clear what constitutes interaction. If it is known to me that someone is offering money for receipts that show a vote for a particular candidate, does the fact that the knowledge has reached me (by whatever means) constitute interaction with the coercer?

Since coercion resistance is generally considered to be a stronger property than receipt freeness, the approach we will take in this paper is to see receipt freeness properties as a subclass of coercion resistance properties. We will assume that receipt freeness deals with a coercer who is concerned only with deducing information about how someone voted from receipts and any public information, but who does not give detailed instructions on how to cast the vote. Coercion resistance, on the other hand, includes dealing with a coercer who gives details not just on which candidate to vote for but also on how to cast the vote.

This understanding of receipt freeness has the advantage that it can be modelled in the same way as coercion resistance. Receipt freeness, on this definition, is equivalent to coercion resistance against a coercer who can specify which candidate the voter should choose, but cannot specify how the voter should make the choice. If the voting process is deterministic (as it is, for example, in Prêt à Voter), then these two notions will coincide, but if it is non-deterministic (as, for example, in ThreeBallot) then they might not.

II. Modelling in CSP

A. CSP background

CSP describes systems in terms of *processes*, which interact by means of synchronising on *events*. The set of all events that a process P can engage in is called its *alphabet*, written αP. Events can be structured, for example $vote.i.v$ may represent a voter i casting a vote v, or $c.v$ represents value v passing on channel c. The set $\{|c|\}$ denotes the set of all events of the form $c.v$. The set of all events is denoted Σ.

The CSP language is used to describe processes. $Stop_A$ represents the process with alphabet A that cannot engage in any events; A may be omitted when it is clear from the context. $Chaos(A)$ is the process with alphabet A that at any stage of execution can nondeterministically perform or refuse to perform events from A. Process $a \rightarrow P$ initially performs a, and subsequently behaves as P. The input process $c?x : S \rightarrow P(x)$ can receive a value $x \in S$ on channel c, and then behave as $P(x)$. The set S can also be expressed as a predicate. The output process $c!v \rightarrow P$ outputs value v along channel c, and then behaves as P. $\bigsqcap_i P_i$ is an internal choice: the process decides which of the P_i to behave as. It also admits a binary form $P \sqcap Q$. $P \Box Q$ offers an external choice: it is prepared to behave as either P or Q, and the choice is resolved by its environment. $\big\|_i P_i$ is the 'alphabet parallel' composition of the P_i. Occurrence of any event a requires the synchronous participation of all processes which have a in their alphabet. This operator also has a binary form $P \parallel Q$. In the 'interface parallel' composition $P \underset{X}{\parallel} Q$, P and Q must synchronise on all events in X, but perform other events independently. $P \setminus A$ behaves as P but with all events from the set A hidden and performed internally. For an event mapping f, the process $f(P)$ behaves as P but performs $f(a)$ whenever P would perform a. Process definitions take the form $N = P$, and can be recursive (i.e. the definition of P contains N). N can also be parameterised.

CSP has several semantic models, which are appropriate for capturing different aspects of process behaviour. In this paper we use the Failures/Divergences model [14], [15], since that model treats non-determinism, and divergence-freedom. In the Failures/Divergences model, a process is associated with two sets F and D, which are respectively the failures and divergences of P. These are understood as observations of possible executions of the process P, in terms of the events from αP that it can engage in.

Divergences of a process are sequences of events which lead to an infinite sequence of internal events. The set *divergences*(P) is the set of all possible divergences for P. In this paper we are generally dealing with divergence-free processes: those for which the set *divergences*(P) is empty.

Stable failures are pairs (tr, X) consisting of a trace $tr \in (\alpha P)^*$ and a set $X \subseteq \alpha P$. This describes P performing the sequence of events tr and then refusing to engage in any of the events in X and being unable to make further internal progress. The set *failures*(P) consists of the stable failures of P, together with pairs (tr, X) for which tr is a divergence. Since we will be dealing with divergence-free processes, *failures*(P) will be the stable failures of P.

If P and Q have the same divergences and failures, then we write $P =_{FD} Q$.

Definition 7 (Deterministic): A process P is *deterministic* if

$$\forall\, tr, X, a \,.\, (tr, X) \in \mathit{failures}(P) \wedge a \in X \\ \Rightarrow (tr \frown \langle a \rangle, \emptyset) \notin \mathit{failures}(P)$$

In this paper we will use failures-divergences refinement, defined as follows:

Definition 8 (Refinement):

$$P \sqsubseteq_{FD} Q \quad \widehat{=} \quad \mathit{failures}(Q) \subseteq \mathit{failures}(P) \\ \wedge\ \mathit{divergences}(Q) \subseteq \mathit{divergences}(P)$$

We may also write $Q \sqsupseteq_{FD} P$ for $P \sqsubseteq_{FD} Q$. Observe that if $P \sqsubseteq_{FD} Q$ and P is divergence-free, then Q is divergence-free. Thus we use failures-divergences refinement throughout to ensure that the refinements we consider for processes are divergence-free.

The particular view of $P \sqsubseteq_{FD} Q$ that we use through this paper is that any behaviour of Q is consistent with, or allowed by, P. In other words, an observer who sees Q cannot tell that it is not P.

B. Voting systems

Throughout this section, we shall assume that voting systems are modelled as follows. The system as a whole is modelled by a process *SYSTEM*; this will be responsible for receiving votes, publishing receipts, tallying, publishing audit data, and whatever else the system in question may need to do.

Voters will interact with the system by being placed in parallel with it. We will model voter behaviour by a process *VOTER*(i, c), which represents the most general behaviour of a voter with ID i who chooses to vote for candidate c.

Preferential voting systems allow voters to rank the candidates, rather than asking them to choose one candidate. The framework presented here is expressive enough to allow for this: c would be the vote in whatever form it might take, rather than necessarily being a specific candidate; or, in other words, each possible ranking would effectively be treated as a separate 'candidate'. However, for clarity of exposition, we will continue to talk in terms of votes for particular candidates.

We will consider coercion resistance and receipt freeness from the perspective of an arbitrarily chosen voter, to whom we will give the name of Zara and the ID of 0. Thus, roughly speaking, we will want to know whether a coercer can distinguish *SYSTEM* $\parallel$ *VOTER*$(0, c)$ from *SYSTEM* $\parallel$ *VOTER*$(0, c')$. In the first case, the target voter casts a vote for c, and in the second case, he casts a vote for c'.

However, we start by observing that no voting system can be coercion resistant from voter 0's perspective if every other voter is under the complete control of the coercer. The coercer will know what the tally should be without voter 0's vote, and so he will be able to establish how voter 0 voted by seeing how the tally has changed. For this reason, we will need to assume that there is at least one other voter who lies outside the control of the coercer. Since we will need to reason about this voter, we will identify him by the name of Juan and the ID of 1. This approach is also taken in the formalisations of coercion resistance and receipt freeness given in [2].

The idea will be that Juan will cover Zara's tracks. For example, consider the case where the coercer instructs Zara to vote for Alice. Coercion resistance will mean that the coercer is unable to distinguish between, on the one hand, Zara's compliance by voting for Alice and Juan's voting for Bob, and, on the other hand, Zara's disobedience by voting for Bob and Juan's voting for Alice. The underlying assumption in this example, then, is that there is at least one voter (whom we will call Juan) who, as far as the coercer is concerned, might or might not vote for Alice, but who in fact does so. As long as at least one voter casts a vote for Alice but is not known by the coercer to have done so, then Zara's non-compliance will be masked. The precise masking behaviour will vary according to the voting system and the model of coercion resistance; we will return later to the question of how reasonable this assumption is for different systems and models.

We now give a semi-formal definition of coercion resistance within this framework. The definition will capture all of the important aspects, but will leave ambiguities in several places; these ambiguities will become parameters to the formal CSP definition to allow us to construct formal definitions of a wide range of notions of coercion resistance and receipt freeness.

Definition 9 (Semi-formal): Suppose that Zara wants to vote for candidate c. Whatever instructions the coercer gives to Zara, there will be a behaviour Z of Zara that casts a vote for c, and two behaviours J_1 and J_2 of Juan, such that, from the coercer's view of the system, Zara's behaving according to Z and Juan's behaving according to J_1 will be consistent with Zara's complying with the coercer's instructions and Juan's behaving according to J_2.

This definition implicitly gives us three important parameters that can be altered to model different notions of coercion resistance:

1) **Coercer's power.** How precise can the coercer's instructions be? For forms where the coercer does not interact with the voter, the coercer will be able to specify a candidate or a set of candidates for whom the voter may vote, but not how the voter must do so, whereas with general coercion resistance properties the coercer may be able to specify precisely how the vote is to be cast.
2) **Abstraction.** How much of the system behaviour can the coercer see, and how much is hidden? For instance, can the coercer see whether Zara has turned up to the polling station? Much will depend on what behaviour we hide from the coercer, and what behaviour we leave.
3) **Abstention.** Might Zara want to abstain? Might the coercer try to force Zara to abstain?

We are now ready to state the formal definition. We start with some assumptions on the model of the system and the model of a general voter.

Definition 10 (Candidates): We denote the set of all candidates by *CANDIDATES*. This set includes the special value abs; a voter who 'chooses' the candidate abs chooses to abstain from voting.

Assumption 1 (System model): The system is modelled by a process *SYSTEM*, and the most general behaviour of an individual voter with ID i who chooses to vote for candidate c is modelled by a process $VOTER(i,c)$. Voter behaviour is also defined for a set of candidates: the most general behaviour of a voter who chooses non-deterministically from the set $CANDS \neq \emptyset$ of candidates is defined by

$$VOTER(i, CANDS) = \bigsqcap_{c \in CANDS} VOTER(i,c)$$

These processes must meet the conditions set out in Figure 1.

One consequence of these assumptions is that voter behaviour and overall system behaviour are both finitary. This rules out, for instance, unbounded auditing of ballot papers in a system like Prêt à Voter [16], or unbounded re-voting in a system like Helios [17]. This is not unreasonable, since in practice polling closes at a fixed time, meaning that systems and voters must eventually terminate their interaction.

What is more important from a technical point of view is that it eliminates the possibility of divergence in any of the processes involved in the model. When we consider the coercer's view of the system, we will abstract away all of the events that the coercer cannot see; if unbounded sequences of such events were allowed, then the abstraction would introduce divergence. By ensuring that every process is divergence free, we will be able to analyze the model in stable failures without concerning ourselves with divergence.

Definition 11 (Coercer's control): We use 'H' to denote the set of events hidden from the coercer's view. The only restriction on this set is that $\{open, close\} \cap H = \emptyset$; in other words, the coercer must be able to see the opening and closing of the election.

Definition 12 (Candidates and abstentions): The set of all candidates under consideration is denoted by 'C'. This will denote all the candidates for whom Zara may wish to vote, and all of the candidates for whom the coercer may wish her to vote. Typically we will have either $C = CANDIDATES \setminus \{\mathsf{abs}\}$, if we do not want to consider abstentions, or $C = CANDIDATES$ if we do.

We now define the set of all instructions the coercer might give Zara. Instructions will come in the form of a process whose behaviour Zara must mimic; for compliance to be possible, the process must be a refinement of $VOTER(0,C)$, Zara's most general behaviour.

Definition 13: We use 'I' to denote the set of instructions that the coercer might give Zara. It must be a subset of the set of all possible instructions that the coercer could give Zara, with the set C of candidates under consideration; in other words:

$$I \subseteq \{P \mid P \sqsupseteq_{FD} VOTER(0,C)\}$$

Definition 14 (Coercion resistance [CR)]): Suppose that we are given some system model *SYSTEM* (with associated voter model $VOTER(i,c)$).

Figure 1 Assumptions on the system model

$$SYSTEM \setminus (\Sigma \setminus \{open, close\}) =_{FD} open \rightarrow close \rightarrow Stop$$

$$VOTER(i, CANDIDATES) \setminus (\Sigma \setminus \{open, close\}) =_{FD} open \rightarrow close \rightarrow Stop$$

$$SYSTEM \parallel (\mathop{\Big\|}_{i \in IDS} VOTER(i, CANDIDATES)) \setminus (\Sigma \setminus \{open, close\}) =_{FD} open \rightarrow close \rightarrow Stop$$

We say that *SYSTEM* meets $CR(I, C, H)$, with

$$I \subseteq \{P \mid P \sqsupseteq_{FD} VOTER(0, C)\}$$
$$C \subseteq CANDIDATES$$
$$H \subseteq \Sigma \setminus \{open, close\}$$

if[1], for all $c \in C$ and $Z_x \in I$, there exist some $Z_c \sqsupseteq_{FD} VOTER(0, c)$ and $J_x \sqsupseteq_{FD} J$ such that

$$\mathcal{L}_H(SYSTEM \parallel Z_x \parallel J) \sqsubseteq_{FD} \mathcal{L}_H(SYSTEM \parallel Z_c \parallel J_x)$$

In this definition, J is a shorthand for $VOTER(1, C)$, Juan's most general behaviour. The set I represents the set of processes that the coercer is able to choose from when giving instructions to Zara; we must have $I \subseteq \{P \mid P \sqsupseteq_{FD} VOTER(0, C)\}$ if Zara is to be able to comply. The second parameter, C, determines the set of candidates under consideration; in particular, the flavour of coercion resistance will change if this contains the special abs candidate. If $\mathsf{abs} \in C$, then Zara must be able to abstain if she so wishes, and the coercer may try to force her to abstain.

The coercer's view is controlled by the third parameter, H. The $\mathcal{L}_H$ function is *lazy abstraction*, and is defined in [14]; it provides a mechanism for masking all of the events (in traces and in refusals) from the hidden set H. It is defined as $\mathcal{L}_H(P) = (P \underset{H}{\parallel} Chaos(H)) \setminus H$. Essentially, by applying lazy abstraction over the set H, we ensure that events from the set H are invisible, so that the coercer can neither see such events nor see the refusal to engage in such events.

What Definition 14 states, then, is that whatever candidate c Zara wishes to vote for, and whatever instructions Z_x the coercer might give her from the set I, there is some possible behaviour Z_c of hers that casts a vote for c, and some possible behaviour J_x of Juan, such that, when we abstract away the set of all hidden events H, any behaviour of the system when Zara acts as Z_c and Juan acts as J_x is also a possible behaviour of the system when Zara acts as instructed by the coercer.

An alternative definition replaces the refinement relation with equality:

Definition 15 (Coercion resistance [CR]):* Suppose that we are given some system model *SYSTEM* (with associated voter model $VOTER(i, c)$).

We say that *SYSTEM* meets $CR^*(I, C, H)$, with

$$I \subseteq \{P \mid P \sqsupseteq_{FD} VOTER(0, C)\}$$
$$C \subseteq CANDIDATES$$
$$H \subseteq \Sigma \setminus \{open, close\}$$

if, for all $c \in C$ and $Z_x \in I$, there exist some $Z_c \sqsupseteq_{FD} VOTER(0, c)$, $J_x \sqsupseteq_{FD} J$, and $J_c \sqsupseteq_{FD} VOTER(1, c)$ such that

$$\mathcal{L}_H(SYSTEM \parallel Z_x \parallel J_c) =_{FD} \mathcal{L}_H(SYSTEM \parallel Z_c \parallel J_x)$$

The difference between Definitions 14 and 15 is that in the former, the question is whether some strategy of Zara's is sufficient to allow her to vote according to her own wishes whilst claiming plausibly to have obeyed the coercer, whereas in the latter, the question is whether *every* observation that the coercer might make of a compliant voter is also possible for a voter voting for c. Definition 15 is stronger than Definition 14 since equality implies refinement.

For most voting systems there will be no difference; but we will see in Section IV-B1 that this is not always the case.

Some of the informal definitions, such as that of Juels, Catalano and Jakobsson and that of Okamoto, are phrased more in line with Definition 14; some others, such as that of Delaune, Kremer and Ryan, and that of Benaloh and Tuinstra, seem to call for Definition 15. Either approach is defensible; and since the purpose of this paper is to give the flexibility to formalize as many different notions as possible, we give both definitions.

The line we will adopt here is to use Definition 14 for the bulk of our work, to illustrate how the definition can be applied. Similar results hold for Definition 15.

III. Definitions of coercion resistance

In this section, we will give formal definitions within our framework of several different informal definitions of coercion resistance and receipt freeness, including some of those found in Section I-A. In each case, we will give the definition of the set I of instructions that the coercer can give. This set will be defined in terms of C, the set of candidates under consideration. We will then give a useful result that allows us to compare definitions; this will enable

[1] We have stated Definition 14 in terms of refinement in the failures-divergences model, because we need to take care throughout our modelling that processes never diverge. However, our assumptions on *SYSTEM* and $VOTER(0, CANDIDATES)$ mean that neither side of the refinement can in fact diverge, so, in practice, checking refinement in the stable failures model is sufficient here.

us to set up a hierarchy of definitions of coercion resistance and receipt freeness.

Since the definitions are in terms of the set I of instructions, they can apply equally to CR and to CR^*.

A. Formal definitions

For convenience, we will attach a superscript of 'abs' whenever the definition includes the special abstention candidate. The definitions below are given in their undecorated form, as if abstentions were not allowed; but later we will use the decorated forms of some of these definitions when we want to consider abstentions.

One notion of coercion resistance that is not given a formal definition below is that of resistance to *randomization attacks*, in which the coercer attempts to force Zara to vote randomly. This type of attack can occur in a system like Prêt à Voter, where the coercer can insist that Zara bring back a receipt with a cross in the top box, without the coercer knowing which candidate the top box will represent. The formal definition of such attacks varies according to the system in question, so we cannot give a general definition, but we will discuss randomization attacks further in Section IV-A.

We have not yet formalized the notion of receipt freeness, but all the definitions below may be considered as general coercion resistance properties (of which receipt freeness properties are a subclass). We will return later to which coercion resistance properties are also receipt freeness properties.

We start with the definition of a general kind of receipt freeness property, in the context of a voter who wishes to deceive the coercer where possible.

Definition 16 (Receipt absence): Our informal definition of receipt absence allows the coercer to specify the content of the vote, but not how to cast the vote. In its most general form, the coercer may specify any non-empty subset X of candidates, and require the voter to cast the vote for a candidate from X. The set of instructions that the coercer may give, then, is

$$I_{RFGEN} = \{VOTER(0, X) \mid X \subseteq C \wedge X \neq \emptyset\}$$

We shall shortly give some results that enable us to say when one definition is stronger than another (in the sense that any system that meets the stronger property also meets the weaker property).

Definition 17 (Okamoto): The Okamoto definition (Definition 1) is captured by

$$I_{OK} = \{VOTER(0, c) \mid c \in C\}$$

Essentially, the coercer may specify a candidate to vote for, but may not specify how the voter is to cast it.

This turns out (Proposition 1) to be equivalent to I_{RFGEN}.

Definition 18 (Benaloh/Tuinstra): We here give the formal definition of coercion resistance for the second interpretation of Definition 4. This holds when a voter aiming to deceive the coercer can avoid leaking information about how the vote was cast.

The Benaloh/Tuinstra definition is captured by

$$I_{BT} = \{P \mid P \sqsupseteq VOTER(0, c) \wedge c \in C\}$$

This is stronger than the Okamoto definition. Here, the coercer can require specific evidence that Zara has complied with specific instructions not just on voting for c but on voting for c in a particular way.

Definition 19 (Delaune/Kremer/Ryan): Definition 2 says that a system is coercion resistant if the coercer cannot tell whether a coerced voter has behaved as instructed or voted a different way. This leaves open the question of what possible instructions the coercer may give, but it appears that in their model a coercer's instructions must always be instructions to vote for a particular candidate, possibly in a specific way. The formal definition within our framework is then the same as the Benaloh/Tuinstra definition: the coercer can choose any candidate, then give instructions in the form of any refinement of the process that always casts a vote for that candidate. Note that Delaune, Kremer and Ryan use observational equivalence, so CR^* will be the appropriate definition corresponding to their definition.

Definition 20 (Chaum): We argued in Section I-A that it is reasonable to take Chaum's definition as equivalent to that of Benaloh/Tuinstra. The formal definition is then also the same:

$$I_{CHAUM} = I_{BT}$$

Definition 21 (Forced abstention attacks): A forced abstention attack is an attack in which the coercer attempts to force Zara to abstain. Since it makes sense only when abstentions are under consideration, we give the formal definition in its decorated form:

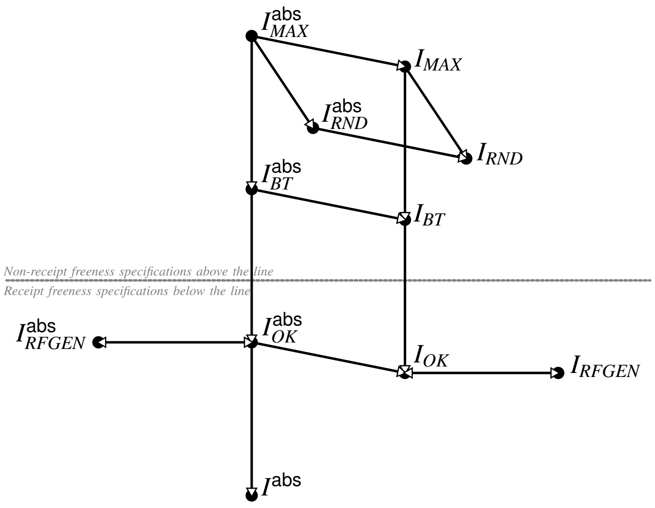
$$I^{\mathsf{abs}} = \{VOTER(0, \mathsf{abs})\}$$

This definition makes the assumption that abstaining is deterministic. If there were a voting system that allowed for non-deterministic ways of abstaining, we could model this as the set of all refinements of $VOTER(0, \mathsf{abs})$. But since abstention usually involves refraining from taking any action, it seems reasonable to model it as deterministic.

Definition 22 (Maximum strength): Our framework finds its strongest possible notion of coercion resistance in the set of all refinements of Zara's most general behaviour, $VOTER(0, C)$. This includes everything covered by Benaloh/Tuinstra, but it also includes randomization attacks, and any other sort of instruction that Zara is able to follow: for instance, an instruction to use the last digit of the ballot serial number to determine which candidate to vote for. When $\mathsf{abs} \in C$, it also includes instructions to abstain, or instructions to participate.

$$I_{MAX} = \{P \mid P \sqsupseteq_{FD} VOTER(0, C)\}$$

Figure 2 Hierarchy of definitions of coercion resistance



B. Relationships between definitions

Some of these definitions are stronger than others. We now state some results that allow us to formalize relations between notions of coercion resistance.

Definition 23 (Dominance): Suppose that I_1 and I_2 are sets of processes. We say that I_1 *dominates* I_2 if

$$\forall P_2 \in I_2 . \exists P_1 \in I_1 . P_2 \sqsubseteq_{FD} P_1$$

Theorem 1 (CR and dominance): Suppose that I_1 dominates I_2, and *SYSTEM* meets $CR(I_1, C, H)$. Then *SYSTEM* also meets $CR(I_2, C, H)$.

Proof: See appendix. ■

Corollary 1 (CR and subset): Suppose that $I_2 \subseteq I_1$, and *SYSTEM* meets $CR(I_1, C, H)$. Then *SYSTEM* also meets $CR(I_2, C, H)$.

Proof: If $I_2 \subseteq I_1$, then I_1 dominates I_2. So Theorem 1 applies, and we conclude that *SYSTEM* meets $CR(I_2, C, H)$. ■

These results allow us to give a hierarchy of definitions. The diagram in Figure 2 shows the relationships among the various formal definitions. An arrow from I_X to I_Y indicates that I_X is stronger than I_Y. In most cases, this is the consequence of a simple subset relationship: $I_Y \subseteq I_X$. The only cases where the relationship is more complex is that of I_{OK} and I_{RFGEN} (with or without abs decoration), where the arrow is double-headed because it turns out that these definitions are equivalent.

Proposition 1 (Okamoto and I_{RFGEN}): For any C and H, the two properties $CR(I_{OK}, C, H)$ and $CR(I_{RFGEN}, C, H)$ are equivalent.

Proof: $I_{OK} \subseteq I_{RFGEN}$, so by Corollary 1, any system that meets I_{RFGEN} meets I_{OK}.

However, any element taken from I_{RFGEN} is of the form $VOTER(0, X)$ for some $X \subseteq C$, with $X \neq \emptyset$. Let us now choose some $x \in X$. Now $VOTER(0, x)$ is in I_{OK}, and $VOTER(0, X) \sqsubseteq_{FD} VOTER(0, x)$. So I_{OK} dominates I_{RFGEN}. Thus, by Theorem 1, we conclude that any system that meets I_{OK} meets I_{RFGEN}.

So the two definitions are equivalent. ■

IV. Examples

In this section, we show how this framework can be used to analyze various different notions of coercion resistance and receipt freeness for various different system models.

A. Simplified Prêt à Voter

The CSP code for a simplified model of Prêt à Voter, running a referendum, is shown in Figure 3. Voters receive a value $b \in \{0, 1\}$ on channel *ballot*, which indicates the ordering of the boxes on the ballot form: a value of 0 indicates that the top box represents 'no', and a value of 1 means that the top box represents 'yes'. They then submit an ID from the finite set *IDS* of all voter IDs, and a value v, which indicates which box they would like to select, with a value of 0 representing the top box; the system returns v to them, and then increments either the 'yes' total or the 'no' total. Finally, when voting closes, the two totals are announced.

Here, and throughout, '$\overline{v}$' represents $1 \oplus v$, where '$\oplus$' is bitwise exclusive-or. (The special candidate abs is treated as invariant under this operation.)

This model intentionally abstracts away the cryptography, the mix servers, and the auditing.

The finitary condition (Assumption 1) for the voter process is clear from the fact that it is non-recursive. The system is finite because on each recursive step the number of people who have voted strictly increases, and cannot exceed $\#IDS$.

Our Prêt à Voter model is rich enough to allow for analysis under various definitions of coercion resistance. We consider several here. Initially, we will not take into account the possibility that Zara or Juan might want to abstain, or the possibility that the coercer might insist on abstention.

Proposition 2 (Okamoto and PaV, no abs): The set of candidates under consideration, when abstentions are not taken into account, is $C_2 = \{0, 1\}$. (Later, we will ask what happens if we include abstentions.)

The Okamoto definition in this setting is encapsulated by

$$I_{OK} = \{VOTER(0, c) \mid c \in C_2\}$$

Suppose that we set $H_{PUB} = \{|ballot|\}$. In other words, the coercer cannot see the ordering of the names on the ballot paper (the *ballot* channel), but can see who arrives to vote (the *arrive* channel) and who ticks which box (the *vote* channel). We use the name 'H_{PUB}' because this models a scenario in which it is made public which voter is associated with each encrypted receipt.

We claim that the simplified Prêt à Voter model meets $CR(I_{OK}, C_2, H_{PUB})$.

Figure 3 A simplified model of Prêt à Voter: defining the system and voter behaviour

$$\begin{aligned}
\mathit{SYSTEM} &= \mathit{open} \rightarrow \mathit{WAITING}(\emptyset, 0, 0) \\
\mathit{WAITING}(\mathit{VOTED}, y, n) &= \mathit{arrive}?\mathit{id} : \mathit{IDS} \setminus \mathit{VOTED} \rightarrow \mathit{VOTING}(\mathit{VOTED}, y, n, \mathit{id}) \\
&\quad \Box\; \mathit{close} \rightarrow \mathit{announce}.y.n \rightarrow \mathit{Stop} \\
\mathit{VOTING}(\mathit{VOTED}, y, n, \mathit{id}) &= \bigsqcap_{b \in \{0,1\}} \mathit{ballot}.\mathit{id}!b \rightarrow \mathit{vote}.\mathit{id}?v \rightarrow \mathit{write}!v \\
&\quad \rightarrow \mathit{WAITING}(\mathit{VOTED} \cup \{\mathit{id}\}, y + (v \oplus b), n + (\bar{v} \oplus b))
\end{aligned}$$

$$\begin{aligned}
\mathit{VOTER}(i, c) = \mathit{open} \rightarrow &\text{if } (c \neq \mathsf{abs}) \text{ then} \\
&\quad \mathit{arrive}!i \rightarrow \mathit{ballot}.i?b \rightarrow \mathit{vote}.i.(c \oplus b) \rightarrow \mathit{close} \rightarrow \mathit{Stop} \\
&\text{else} \\
&\quad \mathit{close} \rightarrow \mathit{Stop}
\end{aligned}$$

Proof: We are required to show that, given $c \in C_2$ and $Z_x \in I_{OK}$, there is some $Z_c \sqsupseteq_{FD} \mathit{VOTER}(0, c)$ and $J_x \sqsupseteq_{FD} J$ such that

$$\mathcal{L}_{H_{PUB}}(\mathit{SYSTEM} \parallel Z_x \parallel J) \sqsubseteq_{FD} \mathcal{L}_{H_{PUB}}(\mathit{SYSTEM} \parallel Z_c \parallel J_x)$$

We start by stating a useful result.

Lemma 1 (Abstraction and renaming): If f is a renaming function that leaves all events outside H unchanged, and $f^{-1}(f(H)) = H$, then

$$\mathcal{L}_H(f(P)) =_{FD} \mathcal{L}_{f(H)}(P)$$

Proof: See Appendix. ■

Corollary 2 (Abstraction and renaming): If f is a bijective renaming function that leaves all events outside H unchanged, then

$$\mathcal{L}_H(f(P)) =_{FD} \mathcal{L}_H(P)$$

Proof: If f is bijective and leaves events outside H unchanged, then it is also bijective on H, and so $f(H) = H$, and $f^{-1}(f(H)) = H$. Thus, by Lemma 1,

$$\mathcal{L}_H(f(P)) =_{FD} \mathcal{L}_H(P)$$

■

Now we are in a position to prove Proposition 2.

Fix some $c \in C_2$. Evidently the case where $Z_x = \mathit{VOTER}(0, c)$ causes us no trouble: this represents the coercer's attempt to force Zara to vote for the candidate c she already wanted to vote for. In this case, $Z_c = Z_x$ and $J_x = J$ will suffice—or, in other words, Zara obeys the coercer and still casts her vote for her preferred candidate.

Now suppose that $Z_x = \mathit{VOTER}(0, \bar{c})$. Zara wishes to vote for c, but the coercer has instructed her to vote for $\bar{c}$. We set $Z_c = \mathit{VOTER}(0, c)$ and $J_x = \mathit{VOTER}(1, \bar{c})$.

Now we are required to check that

$$\begin{aligned}
&\mathcal{L}_{H_{PUB}}(\mathit{SYSTEM} \parallel \mathit{VOTER}(0, \bar{c}) \parallel J) \\
&\qquad \sqsubseteq_{FD} \\
&\quad \mathcal{L}_{H_{PUB}}(\mathit{SYSTEM} \parallel \mathit{VOTER}(0, c) \parallel \mathit{VOTER}(1, \bar{c})) \qquad (1)
\end{aligned}$$

This can be mechanically checked in FDR with a reasonably small set of IDs, for each of the two values of c. However, it is clear that the relation holds in any case. Neither *VOTER*'s behaviour nor *SYSTEM*'s behaviour depends in any way on the IDs, except for the equality check implicit in the restriction that voters may vote only once. In other words, if π is a permutation on IDs, and f_π is a renaming function that simply changes each ID i to $\pi(i)$:

$$\begin{aligned}
f_\pi(\mathit{arrive}.i) &= \mathit{arrive}.\pi(i) \\
f_\pi(\mathit{ballot}.i) &= \mathit{ballot}.\pi(i) \\
f_\pi(\mathit{vote}.i) &= \mathit{vote}.\pi(i) \\
f_\pi(x) &= x \qquad (x \notin \{|\mathit{arrive}, \mathit{ballot}, \mathit{vote}|\})
\end{aligned}$$

then

$$\begin{aligned}
f_\pi(\mathit{VOTER}(i, \mathit{CANDS})) &= \mathit{VOTER}(\pi(i), \mathit{CANDS}) \\
f_\pi(\mathit{SYSTEM}) &= \mathit{SYSTEM}
\end{aligned}$$

Now let π be the permutation that swaps 0 and 1, and leaves all other IDs unchanged. Then

$$\begin{aligned}
&f_\pi(\mathit{SYSTEM} \parallel \mathit{VOTER}(0, c) \parallel \mathit{VOTER}(1, \bar{c})) \\
&\quad = \mathit{SYSTEM} \parallel f_\pi(\mathit{VOTER}(0, c)) \parallel f_\pi(\mathit{VOTER}(1, \bar{c})) \\
&\quad = \mathit{SYSTEM} \parallel \mathit{VOTER}(1, c) \parallel \mathit{VOTER}(0, \bar{c}) \\
&\quad = \mathit{SYSTEM} \parallel \mathit{VOTER}(0, \bar{c}) \parallel \mathit{VOTER}(1, c)
\end{aligned}$$

In other words, the effect of switching Zara's vote with Juan's vote is only to swap over their IDs wherever they appear.

Applying $f_{\pi^{-1}}$ to both sides, we have that

$$\begin{aligned} &\mathit{SYSTEM} \parallel \mathit{VOTER}(0, c) \parallel \mathit{VOTER}(1, \bar{c}) \\ &\quad = f_{\pi^{-1}}(\mathit{SYSTEM} \parallel \mathit{VOTER}(0, \bar{c}) \parallel \mathit{VOTER}(1, c)) \end{aligned}$$

So Equation 1 now becomes:

$$\begin{aligned} &\mathcal{L}_{H_{PUB}}(\mathit{SYSTEM} \parallel \mathit{VOTER}(0, \bar{c}) \parallel J) \\ &\qquad\qquad \sqsubseteq_{FD} \\ &\quad \mathcal{L}_{H_{PUB}}(f_{\pi^{-1}}(\mathit{SYSTEM} \parallel \mathit{VOTER}(0, \bar{c}) \parallel \mathit{VOTER}(1, c))) \end{aligned}$$

It is clear that if we were to remove the lazy abstraction and the renaming then the right-hand side would be a refinement of the left, simply because $\mathit{VOTER}(1, c)$ is a refinement of J. But the renaming only affects events that are in H_{PUB}, and $f_{\pi^{-1}}$ is bijective, so we can appeal to Corollary 2 to establish the result. ■

We now consider forced randomization attacks. In such an attack, the coercer does not attempt to force the voter to cast a vote for a specific candidate, but 'neutralizes' the vote by forcing the voter to cast a random vote. This type of attack could be employed in, for instance, districts believed to favour the coercer's main rival.

Proposition 3 (Randomization attacks and PaV, no abs*):* To mount a randomization attack under Prêt à Voter, the coercer specifies a particular box to be ticked (for instance, the top box). The coercer has no means of knowing whether this box represents a 'yes' vote or a 'no' vote.

Such an attack is represented in our model by setting

$$\begin{aligned} I_{RND} = \{&\mathit{open} \rightarrow \mathit{arrive}!0 \rightarrow \mathit{ballot}.0?b \rightarrow \\ &\quad \mathit{vote}.i.v \rightarrow \mathit{close} \rightarrow \mathit{Stop} \\ &\mid v \in \{0, 1\}\} \end{aligned}$$

As before, we consider candidates in $C_2 = \{0, 1\}$ and an abstraction of $H_{PUB} = \{|\mathit{ballot}|\}$. The coercer can see which box Zara ticks, but not which candidate the box represents.

We claim that our simplified model of Prêt à Voter does not meet $\mathit{CR}(I_{RND}, C_2, H_{PUB})$.

Proof: We are required to show that, for some $c \in C_2$ and $Z_x \in I_{RND}$, there do not exist $Z_c \sqsupseteq_{FD} \mathit{VOTER}(0, c)$ and $J_x \sqsupseteq_{FD} J$ such that

$$\mathcal{L}_{H_{PUB}}(\mathit{SYSTEM} \parallel Z_x \parallel J) \sqsubseteq_{FD} \mathcal{L}_{H_{PUB}}(\mathit{SYSTEM} \parallel Z_c \parallel J_x)$$

Let us set $c = 0$ and

$$\begin{aligned} Z_x = \mathit{open} \rightarrow \mathit{arrive}!0 &\rightarrow \mathit{ballot}.0?b \rightarrow \\ &\mathit{vote}.0.0 \rightarrow \mathit{close} \rightarrow \mathit{Stop} \end{aligned}$$

In other words, Zara wants to vote 'no', and the coercer insists that she tick the top box.

The first point to observe is that $\mathit{VOTER}(0, c)$ is deterministic, and since the only refinement of a deterministic process in the failures-divergences model is itself, the only possible choice for Z_c is $\mathit{VOTER}(0, c)$.

However, $t = \langle \mathit{open}, \mathit{arrive}.0, \mathit{ballot}.0.1, \mathit{vote}.0.1 \rangle$ is a possible trace of $\mathit{VOTER}(0, c)$. This represents Zara's being given a ballot form with 'yes' at the top, and voting 'no' by ticking the bottom box. *SYSTEM* also has t as a possible trace.

Whatever the choice of J_x, it must allow $t \upharpoonright \alpha(J_x) = \langle \mathit{open} \rangle$. It cannot block any of the other events in the trace, since they lie outside its alphabet.

Therefore, t is a trace of $\mathit{SYSTEM} \parallel Z_c \parallel J_x$. The effect of the abstraction is to hide the *ballot* event, but not block the others. So $t' = \langle \mathit{open}, \mathit{arrive}.0, \mathit{vote}.0.1 \rangle$ is a trace of the right-hand side.

However, t' is not a trace of the left-hand side. The $\mathit{vote}.0.1$ event is in the alphabet of Z_x, so if it were to occur, it would have to appear in a trace of Z_x. But one can see from the definition of Z_x that it never engages in such an event.

So t' is a trace of the right-hand side but not of the left-hand side. The refinement therefore does not hold, and the system does not meet $\mathit{CR}(I_{RND}, C_2, H_{PUB})$. ■

Corollary 3 (I_{MAX} and PaV, no abs*):* It is an immediate corollary of Proposition 3 and Corollary 1 that our simplified Prêt à Voter does not meet $\mathit{CR}(I_{MAX}, C_2, H_{PUB})$. Any set of coercer instructions must be a subset of I_{MAX}, so Corollary 1 tells us that if Prêt à Voter met $\mathit{CR}(I_{MAX}, C_2, H_{PUB})$ then it would meet $\mathit{CR}(I, C_2, H_{PUB})$ for any I. But Proposition 3 shows that it does not meet $\mathit{CR}(I_{RND}, C_2, H_{PUB})$; therefore, it cannot meet $\mathit{CR}(I_{MAX}, C_2, H_{PUB})$.

We now return to the question of abstentions. In what follows, we will use $C_2^{\mathsf{abs}} = C_2 \cup \{\mathsf{abs}\}$, and establish what effect this has on coercion resistance. Including abs has two consequences:

1) Zara may now want to abstain; coercion resistance will imply that she is able to abstain if she wants to, without the coercer knowing that she has not voted according to instructions. If the coercer can force Zara not to abstain, then we have a *forced participation* attack.
2) The coercer may insist on Zara's abstention; coercion resistance will imply that she is able to vote if she wants to, without the coercer knowing that she has not abstained. If the coercer can force Zara to abstain, then we have a *forced abstention* attack.

The model is rich enough to handle these cases independently. However, they are naturally treated together, and we will treat them together here.

Proposition 4 (Okamoto and PaV, C_2^{abs}): The Okamoto definition, with abs included, is modelled by

$$I_{OK}^{\mathsf{abs}} = \{\mathit{VOTER}(0, c) \mid c \in C_2^{\mathsf{abs}}\}$$

By including abs in the set of candidates, we also allow for the possibility that Zara wishes to abstain. We continue to set $H_{PUB} = \{|ballot|\}$, so that the coercer can see all voter actions but cannot see the candidate ordering on the ballot paper.

We claim that our simplified Prêt à Voter model does not meet $CR(I_{OK}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{PUB})$.

Proof: Let us set $c = 0$, and

$$Z_x = VOTER(0, \mathsf{abs}) = open \rightarrow close \rightarrow Stop$$

indicating that Zara wishes to vote 'yes', but the coercer insists that she should abstain.

We now need to show that there do not exist $Z_c \sqsupseteq_{FD} VOTER(0,0)$ and $J_x \sqsupseteq_{FD} J$ such that

$$\mathcal{L}_{H_{PUB}}(SYSTEM \parallel VOTER(0, \mathsf{abs}) \parallel J) \sqsubseteq_{FD} \mathcal{L}_{H_{PUB}}(SYSTEM \parallel Z_c \parallel J_x)$$

As before, $VOTER(0,0)$ is deterministic, so we must have $Z_c = VOTER(0,0)$. But $VOTER(0,0)$ has $t = \langle open, arrive.0\rangle$ as a trace. This is also a trace of $SYSTEM$, and $t \upharpoonright \alpha(J) = \langle open\rangle$, which is a trace of all refinements of J. Neither of the events in t is abstracted away by H_{PUB}, so t is a trace of the right-hand side.

It cannot, however, be a trace of the left-hand side. The event $arrive.0$ is blocked from occurring by $VOTER(0, \mathsf{abs})$, which never engages in this event. So Prêt à Voter does not meet $CR(I_{OK}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{PUB})$. ■

This is what we should have expected. If the coercer can see voters arriving, or can see voters' receipts, or anything that includes the voter's ID, then there is no hope of resistance to forced abstention attacks.

Corollary 4 (I_{MAX}^{abs} and PaV, C_2^{abs}): The strongest definition, with abs included, is modelled by

$$I_{MAX}^{\mathsf{abs}} = \{P \mid P \sqsupseteq_{FD} VOTER(0,c) \mid c \in C_2^{\mathsf{abs}}\}$$

We claim that our simplified Prêt à Voter model does not meet $CR(I_{MAX}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{PUB})$.

Proof: This is an immediate consequence of Proposition 4 and Corollary 1. Since the model does not meet $CR(I_{OK}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{PUB})$, and $I_{OK}^{\mathsf{abs}} \subseteq I_{MAX}^{\mathsf{abs}}$, the model does not meet $CR(I_{MAX}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{PUB})$ either. ■

Our final consideration with our Prêt à Voter model is to ask what happens if we change the level of abstraction, so that the coercer can see fewer events. In particular, we will alter the abstraction so that the coercer can see votes being posted up (on the *write* channel), but not arrivals or vote casting. We will set $H_{SEC} = \{|arrive, ballot, vote|\}$.

Proposition 5 (I_{MAX} and PaV, C_2^{abs}, H_{SEC}): Our simplified Prêt à Voter model meets $CR(I_{MAX}, C_2^{\mathsf{abs}}, H_{SEC})$. In other words, when all events containing voter IDs are abstracted away, our model satisfies the strongest possible definition of coercion resistance in our framework.

Proof: The proof closely parallels that of Proposition 2.

Fix some $c \in C_2^{\mathsf{abs}}$, and some $Z_x \sqsupseteq_{FD} VOTER(0, C_2^{\mathsf{abs}})$. Now we set $Z_c = VOTER(0,c)$, and $J_x = f_\pi(Z_x)$. The function f_π is exactly as in Proposition 2: it simply swaps all occurrences of Zara's ID for Juan's, and vice versa.

We are required to show that

$$\mathcal{L}_{H_{SEC}}(SYSTEM \parallel Z_x \parallel J) \sqsubseteq_{FD} \mathcal{L}_{H_{SEC}}(SYSTEM \parallel Z_c \parallel f_\pi(Z_x))$$

Corollary 2 and the bijective nature of f_π tell us that

$$\begin{aligned}
&\mathcal{L}_{H_{SEC}}(SYSTEM \parallel Z_c \parallel f_\pi(Z_x)) \\
&\quad = \mathcal{L}_{H_{SEC}}(f_{\pi^{-1}}(SYSTEM \parallel Z_c \parallel f_\pi(Z_x))) \\
&\quad = \mathcal{L}_{H_{SEC}}(SYSTEM \parallel f_{\pi^{-1}}(Z_c) \parallel Z_x) \\
&\quad = \mathcal{L}_{H_{SEC}}(SYSTEM \parallel Z_x \parallel f_{\pi^{-1}}(Z_c))
\end{aligned}$$

But

$$\begin{aligned}
f_{\pi^{-1}}(Z_c) &= f_{\pi^{-1}}(VOTER(0,c)) \\
&= VOTER(1,c) \\
&\sqsupseteq_{FD} VOTER(1, C_2^{\mathsf{abs}}) \\
&= J
\end{aligned}$$

and so, by the monotonicity of the standard CSP operators,

$$\mathcal{L}_{H_{SEC}}(SYSTEM \parallel Z_x \parallel J) \sqsubseteq_{FD} \mathcal{L}_{H_{SEC}}(SYSTEM \parallel Z_c \parallel f_\pi(Z_x))$$

■

It is evident from this one example that the framework we have constructed is able to handle a wide variety of notions of coercion resistance, by varying the values of I, C and H. A summary of results is shown in Table I.

B. Further Examples

Two further examples illustrate the differences between various types of coercion resistance. 'Two-receipt' shows the difference between the definitions of Okamoto (where it holds) and Benaloh/Tuinstra (where it does not hold). 'Opt-receipt' shows the difference between the two characterisations of coercion resistance, CR and CR^*.

1) Two-receipt: This system allows voters to obtain a receipt containing two names (listed in arbitrary order), consisting of the candidate who received the vote and one other candidate of the voter's choice. The intention is that the inclusion of an alternative name on the receipt allows the voter to mask who received her vote. This system is described in Figure 4. The running tally is maintained by a bag (multiset) t containing the votes cast so far. The receipt is modelled as $write.id.\{c, c'\}$. Then $SYSTEM2$ meets the property $CR(I_{OK}, \{c_1, c_2, c_3\}, \{|vote, dummy|\})$. A voter instructed to vote for c' can vote for c in a way consistent with a vote for c'. The initial parameters of $SYSTEM2$ are the empty set (of voters who have so far voted), and the empty bag (of votes cast so far).

Conversely, $SYSTEM2$ does not meet the Benaloh and Tuinstra characterisation as captured by the property

Table I Summary of results for simplified Prêt à Voter model

Definition	**Abstentions?**	**Invisible events**	**Formalism**	**Met by Prêt à Voter?**
Okamoto	No	$\{\|ballot\|\}$	$CR(I_{OK}, C_2, H_{PUB})$	Yes
Randomization	No	$\{\|ballot\|\}$	$CR(I_{RND}, C_2, H_{PUB})$	No
Strongest possible	No	$\{\|ballot\|\}$	$CR(I_{MAX}, C_2, H_{PUB})$	No
Okamoto / forced abstention	Yes	$\{\|ballot\|\}$	$CR(I^{\mathsf{abs}}_{OK}, C^{\mathsf{abs}}_2, H_{PUB})$	No
Strongest possible	Yes	$\{\|ballot\|\}$	$CR(I^{\mathsf{abs}}_{MAX}, C^{\mathsf{abs}}_2, H_{PUB})$	No
Strongest possible	Yes	$\{\|arrive, ballot, vote\|\}$	$CR(I^{\mathsf{abs}}_{MAX}, C^{\mathsf{abs}}_2, H_{SEC})$	Yes

Figure 4 A model of Two-receipt

$$SYSTEM2 = open \rightarrow WAITING2(\emptyset, \lbag\, \rbag)$$

$$WAITING2(VOTED, t) = arrive?id : IDS \setminus VOTED \rightarrow VOTING2(VOTED, t, id)$$

$$\Box\ close \rightarrow announce!t \rightarrow Stop$$

$$VOTING2(VOTED, t, id) = vote.id?v \rightarrow dummy.id?v' : \{v' \neq v\} \rightarrow write.id!\{v, v'\} \rightarrow WAITING2(VOTED \cup \{id\}, t \uplus \lbag v \rbag)$$

$$VOTER2(id, c) = open \rightarrow arrive!id \rightarrow vote.id.c \rightarrow \bigsqcap_{c' \neq c} dummy.id.c' \rightarrow write.id.\{c, c'\} \rightarrow close \rightarrow Stop$$

$CR(I_{BT}, \{c_1, c_2, c_3\}, \{|vote, dummy|\})$. To see this, consider the case where the voter wishes to vote for $c = c_3$, and the coercer's instruction is $Z_x = vote.0.c_1 \rightarrow dummy.0.c_2 \rightarrow write.0.\{c_1, c_2\} \rightarrow close \rightarrow Stop$. There is no $Z_{c_3} \sqsupseteq_{FD} VOTER(0, c_3)$ and $J_x \sqsupseteq_{FD} J$ which satisfy the conditions of Definition 14 (coercion resistance). In particular, for any Z_{c_3} and J_x the refinement condition will not hold:

$$\mathcal{L}_H(SYSTEM2 \parallel Z_x \parallel J) \not\sqsubseteq_{FD} \mathcal{L}_H(SYSTEM2 \parallel Z_{c_3} \parallel J_x)$$

The right hand side must include a trace containing $write.0.\{c_3, c'\}$ for some c', since this will be generated by Z_{c_3}; but no such trace is possible for the left-hand side.

This consistent with our expectations. A voter is able to vote for her preferred candidate c in a way consistent with a vote for c', as required by Okamoto's definition. On the other hand, if the coercer can require a vote to be cast in a particular way, then the voter might not be able to vote in her preferred way consistent with this. Our formal characterisation captures this distinction.

2) Opt-receipt: The following example is attributed to Ron Rivest [18]. On accepting a vote, the system chooses whether or not to offer a receipt. If offered, the voter chooses whether or not to accept the receipt. Hence the voter might obtain a receipt of exactly how they voted. However, they can also vote for their preferred candidate in a way consistent with any instructions a coercer might give them, by declining any receipt, and claiming that the system did not offer one.

Our model of the Opt-receipt system is given in Figure 5. It meets $CR(I_{MAX}, C, H_{OPT})$, where we set $H_{OPT} = \{|vote, noreceipt, offerreceipt, yes, no|\}$. This captures the sense that the voter has a strategy for voting without production of a receipt, and this is indeed true for $SYSTEM3$.

However, the system does not meet $CR^*(I_{MAX}, C, H_{OPT})$, or even $CR(I_{OK}, C, H_{OPT})$. In particular, if $Z_x = VOTER(0, c')$ and $c \neq c'$ then there is no $Z_c \sqsupseteq_{FD} VOTER(0, c)$, $J_x \sqsupseteq J$ and $J_c \sqsupseteq_{FD} VOTER(1, c)$ such that

$$\mathcal{L}_{H_{OPT}}(SYSTEM3 \parallel Z_x \parallel J_c) \sqsubseteq_{FD} \mathcal{L}_{H_{OPT}}(SYSTEM3 \parallel Z_c \parallel J_x)$$

The reason this equivalence cannot hold is that the event $write.0.c'$ is in a possible trace of the left-hand side, so the equivalence condition requires that the right-hand process must allow the same trace. However, Z_c cannot perform $write.0.c'$ (since $c \neq c'$), and nor can J_x, so the trace is not possible for the right-hand side.

This example thus highlights the difference between CR, which requires the existence of a coercion resistance strategy for a voter, and CR^*, which requires that information about the vote should not leak whatever the voter does.

3) Num-receipt: In this example, when casting a vote, the voter also provides an arbitrary number to be associated with that vote. The system posts all of the votes and their associated numbers onto a public bulletin board. The voter does not obtain any separate receipt.

Such a system allows a coercer to provide a voter with a value to use that is unlikely to be chosen by any other voter. The coercer requires that value to appear on the bulletin board against the candidate of the coercer's choice.

The system fails coercion-resistance in the case where a coercer can communicate with the voter before the end of the

Figure 5 A model of Opt-receipt

$$\begin{aligned}
\mathit{SYSTEM}3 &= \mathit{open} \to \mathit{WAITING}3(\emptyset, \lbag\, \rbag) \\
\mathit{WAITING}3(\mathit{VOTED}, t) &= \mathit{arrive}?\mathit{id} : \mathit{IDS} \setminus \mathit{VOTED} \to \mathit{VOTING}3(\mathit{VOTED}, t, \mathit{id}) \\
&\quad \Box\ \mathit{close} \to \mathit{announce}!t \to \mathit{Stop} \\
\mathit{VOTING}3(\mathit{VOTED}, t, \mathit{id}) &= \mathit{vote.id}?v \to \\
&\qquad \mathit{noreceipt.id} \to \mathit{WAITING}3(\mathit{VOTED} \cup \{\mathit{id}\}, t \uplus \lbag v \rbag) \\
&\qquad \sqcap\ \mathit{offerreceipt.id} \to (\mathit{yes.id} \to \mathit{write.id}!v \to \mathit{WAITING}3(\mathit{VOTED} \cup \{\mathit{id}\}, t \uplus \lbag v \rbag)) \\
&\qquad\qquad\qquad\qquad \Box\ \mathit{no.id} \to \mathit{SYSTEM}3(\mathit{VOTED} \cup \{\mathit{id}\}, t \uplus \lbag v \rbag))
\end{aligned}$$

$$\begin{aligned}
\mathit{VOTER}3(\mathit{id}, c) = \mathit{open} \to \mathit{arrive}!\mathit{id} \to \mathit{vote.id.c} \to (&\mathit{noreceipt.id} \to \mathit{close} \to \mathit{Stop} \\
&\Box\ \mathit{offerreceipt.id} \to (\mathit{yes.id} \to \mathit{write.id.c} \to \mathit{close} \to \mathit{Stop} \\
&\qquad\qquad\qquad \sqcap\ \mathit{no.id} \to \mathit{close} \to \mathit{Stop}))
\end{aligned}$$

voting phase. However, in the case of receipt-freeness, where a coercer cannot communicate with the voter until after the election, the voter does not obtain evidence of having voted for the coercer's preferred candidate: she can simply claim that any appropriate vote from the bulletin board was hers. Hence the system satisfies CR and CR^* for I_{OK}, but not for I_{BT}

4) Opt-num-receipt: Finally, we consider a cross between Opt-receipt and Num-receipt. In this example, the voter provides a number of their choice when they cast their vote, as in Num-receipt. The system then nondeterministically chooses whether or not to offer the voter the option of having their vote and chosen value posted on the bulletin board. The voter can accept or reject the offer.

This example provides receipt-freeness for the same reasons as Num-receipt above. Furthermore, it is possible for the voter to vote for their preferred candidate in a way consistent with having followed the coercer's instructions, by claiming that the system did not offer the option of posting the vote on the bulletin board. Thus this example also meets $CR(I_{BT})$. However, it does not meet $CR^*(I_{BT})$: it is possible for the voter to demonstrate she has complied with the coercer's instructions.

The four examples given in this section were presented to highlight the differences between the different combinations of the CR and CR^* properties with I_{OK} and I_{BT}. A summary of the properties they meet is presented in Figure 6.

V. Receipt freeness

As discussed in Section I-A, we consider receipt freeness as a subclass of coercion resistance, in which a coercer does not give instructions on how to vote, and is concerned only with information leakage about how the vote was cast. Appropriate instantiations of I will thus be dominated by I_{OK}. Each definition of coercion resistance gives rise to a version of receipt freeness.

Definition 24 (Receipt freeness [RF)]): A system meets $RF(I, C, H)$ if it meets $CR(I, C, H)$ and I_{OK} dominates I.

Definition 25 (Receipt freeness [RF)]):* A system meets $RF^*(I, C, H)$ if it meets $CR^*(I, C, H)$ and I_{OK} dominates I.

The following lemma establishes a sufficient condition for both forms of receipt freeness:

Lemma 2: If, for any $x, c \in C$,

$$\begin{aligned}
&\mathcal{L}_H(\mathit{SYSTEM} \parallel \mathit{VOTER}(0, x) \parallel \mathit{VOTER}(1, c)) \\
&\qquad =_{FD} \mathcal{L}_H(\mathit{SYSTEM} \parallel \mathit{VOTER}(0, c) \parallel \mathit{VOTER}(1, x))
\end{aligned}$$

then SYSTEM meets $RF(I_{OK}, C, H)$ and $RF^*(I_{OK}, C, H)$.

A. Deterministic voters

In many voting systems, the process of casting a vote for a particular candidate is deterministic. For such systems , it follows that $I_{OK} = I_{BT}$: once the instruction to vote for c is given, no additional instructions are possible and the ability to interact with the voter gives the coercer no additional advantage.

However, note that I_{MAX} might not be dominated by I_{BT}, and so even in the case of deterministic voting, the coercion resistance property $CR(I_{MAX}, C, H)$ or $CR^*(I_{MAX}, C, H)$ does not always collapse to receipt freeness. Indeed, Prêt à Voter provides an example of this: it allows randomisation attacks even though the voting process is deterministic, and the system is receipt-free.

VI. Conclusion

Definitions of coercion resistance and receipt freeness in the literature are often imprecise, and even where formal definitions are given, two definitions from two different sources rarely coincide.

In this paper, we have constructed a framework for modelling and analyzing coercion resistance and receipt freeness properties in CSP. We have shown how several of the most

Figure 6 Properties satisfied by example voting protocols.

	$CR(I_{OK}, C, H)$	$CR(I_{BT}, C, H)$	$CR^*(I_{OK}, C, H)$	$CR^*(I_{BT}, C, H)$
Two-receipt	✓	×	×	×
Opt-receipt	✓	✓	×	×
Num-receipt	✓	×	✓	×
Opt-num-receipt	✓	✓	✓	×
Prêt à Voter	✓	✓	✓	✓

frequently used definitions in the literature can be encapsulated within our setup, and we have given results that allow two different definitions to be compared, and relationships between them discovered. The only other attempt we are aware of to provide a general framework is that of [2]. Their approach provides characterisations of coercion resistance and receipt freeness, as well as privacy. Broadly speaking, they restrict attention only to the forms CR^* and RF^*, and their framework cannot handle randomization attacks (or anything stronger than I_{BT}).

Our framework is rich enough to encompass a broad range of definitions within the same setup, so that, using our framework, one can model a system, and ask precisely which notions of coercion resistance and receipt freeness it satisfies. This has a distinct advantage that one can work with a variety of definitions without translating the model into several different notations.

Acknowledgements

Thanks are due to Chris Culnane, Murat Moran, Sriram Srinivasan and Zhe (Joson) Xia for their helpful comments and discussions. This work was conducted while the first author was a Royal Society/Leverhulme Trust Senior Research Fellow. The research was funded by EPSRC under grant EP/G025797/1.

References

[1] T. Moran and M. Naor, "Receipt-free universally-verifiable voting with everlasting privacy," in *CRYPTO*, ser. Lecture Notes in Computer Science, C. Dwork, Ed., vol. 4117. Springer, 2006, pp. 373–392.

[2] S. Delaune, S. Kremer, and M. Ryan, "Verifying privacy-type properties of electronic voting protocols," *Journal of Computer Security*, vol. 17, no. 4, pp. 435–487, 2009.

[3] T. Okamoto, "Receipt-free electronic voting schemes for large scale elections," in *Security Protocols Workshop*, ser. Lecture Notes in Computer Science, B. Christianson, B. Crispo, T. M. A. Lomas, and M. Roe, Eds., vol. 1361. Springer, 1997, pp. 25–35.

[4] M. Backes, C. Hritcu, and M. Maffei, "Automated verification of remote electronic voting protocols in the applied pi-calculus," in *CSF*. IEEE Computer Society, 2008, pp. 195–209.

[5] R. Küsters and T. Truderung, "An epistemic approach to coercion-resistance for electronic voting protocols," in *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2009, pp. 251–266.

[6] D. Unruh and J. Müller-Quade, "Universally composable incoercibility," in *CRYPTO*, ser. Lecture Notes in Computer Science, T. Rabin, Ed., vol. 6223. Springer, 2010, pp. 411–428.

[7] O. de Marneffe, O. Pereira, and J.-J. Quisquater, "Simulation-based analysis of e2e voting systems," in *VOTE-ID*, ser. Lecture Notes in Computer Science, A. Alkassar and M. Volkamer, Eds., vol. 4896. Springer, 2007, pp. 137–149.

[8] A. Alkassar and M. Volkamer, Eds., *E-Voting and Identity, First International Conference, VOTE-ID 2007, Bochum, Germany, October 4-5, 2007, Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 4896. Springer, 2007.

[9] J. C. Benaloh and D. Tuinstra, "Receipt-free secret-ballot elections (extended abstract)," in *STOC*, 1994, pp. 544–553.

[10] D. Chaum, P. Y. A. Ryan, and S. Schneider, "A practical voter-verifiable election scheme," in *ESORICS*, ser. Lecture Notes in Computer Science, S. D. C. di Vimercati, P. F. Syverson, and D. Gollmann, Eds., vol. 3679. Springer, 2005, pp. 118–139.

[11] D. Chaum, J. V. D. Graaf, P. Y. A. Ryan, and P. L. Vora, "Secret ballot elections with unconditional integrity," 2007, technical report.

[12] R. L. Rivest and W. D. Smith, "Three voting protocols: ThreeBallot, VAV, and Twin," in *Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop*, 2007.

[13] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in *WPES*, V. Atluri, S. D. C. di Vimercati, and R. Dingledine, Eds. ACM, 2005, pp. 61–70.

[14] A. Roscoe, *Theory and Practice of Concurrency*. Prentice-Hall, 1998.

[15] S. Schneider, *Concurrent and Real-time Systems: The CSP approach*. Wiley, 1999.

[16] D. Chaum, P. Y. A. Ryan, and S. A. Schneider, "A Practical Voter-Verifiable Election Scheme," in *10th European Symposium On Research In Computer Security (ESORICS 2005)*, ser. Lecture Notes in Computer Science, vol. 3679. Springer-Verlag, 2005, pp. 118–139.

[17] B. Adida, "Helios: Web-based open-audit voting," in *USENIX Security Symposium*, P. C. van Oorschot, Ed. USENIX Association, 2008, pp. 335–348.

[18] P. Y. A. Ryan, personal communication.

Appendix

Proof of Lemma 1

The lemma states that if f is the identity function on events outside H, and $f^{-1}(f(H)) = H$, then $\mathcal{L}_H(P) = \mathcal{L}_{f(H)}(f(P))$.

Proof:

We will make use of the following three laws of CSP. We write '$f_{\setminus A}$' to denote the projection of f onto domain $\Sigma \setminus A$.

1) If $f^{-1}(f(A)) = A$ and $f_{\setminus A}$ is finite-to-one, then

$$(f_{\setminus A})(P \setminus A) = f(P) \setminus f(A)$$

2) If $f^{-1}(f(A)) = A$ then

$$f(P \underset{A}{\parallel} \mathit{Chaos}(A)) = f(P) \underset{f(A)}{\parallel} \mathit{Chaos}(f(A))$$

3) $f(\mathit{Chaos}(A)) = \mathit{Chaos}(f(A))$

We are now ready to prove the lemma. The conditions on f are that $f_{\setminus H} = id_{\Sigma \setminus H}$, the identity on $\Sigma \setminus H$; and that $f^{-1}(f(H)) = H$. The first condition implies that $f_{\setminus H}$ is finite-to-one (in fact 1-1), so the three laws are all applicable:

$$\begin{aligned}
\mathcal{L}_H(P) &= (P \underset{H}{\parallel} \mathit{Chaos}(H)) \setminus H \\
&= (f_{\setminus H})((P \underset{H}{\parallel} \mathit{Chaos}(H)) \setminus H) \\
&= f(P \underset{H}{\parallel} \mathit{Chaos}(H)) \setminus f(H) \\
&= (f(P) \underset{f(H)}{\parallel} f(\mathit{Chaos}(H))) \setminus f(H) \\
&= (f(P) \underset{f(H)}{\parallel} \mathit{Chaos}(f(H))) \setminus f(H) \\
&= \mathcal{L}_{f(H)}(f(P))
\end{aligned}$$

■

Proof of Theorem 1

The theorem states that if I_1 dominates I_2, and *SYSTEM* meets $\mathit{CR}(I_1, C, H)$, then *SYSTEM* also meets $\mathit{CR}(I_2, C, H)$.

Proof: If *SYSTEM* meets $\mathit{CR}(I_1, C, H)$, then for all $c \in C$ and $Z_{x_1} \in I_1$, there exist some $Z_c \sqsupseteq_{FD} \mathit{VOTER}(0, c)$ and $J_x \sqsupseteq_{FD} J$ such that

$$\mathcal{L}_H(\mathit{SYSTEM} \parallel Z_{x_1} \parallel J) \sqsubseteq_{FD} \mathcal{L}_H(\mathit{SYSTEM} \parallel Z_c \parallel J_x)$$

(Recall that $J = \mathit{VOTER}(1, C)$.)

We must show that this holds for I_2 as well; that is, for all $c \in C$ and $Z_{x_2} \in I_2$, there exist some $Z_c \sqsupseteq_{FD} \mathit{VOTER}(0, c)$ and $J_x \sqsupseteq_{FD} J$ such that

$$\mathcal{L}_H(\mathit{SYSTEM} \parallel Z_{x_2} \parallel J) \sqsubseteq_{FD} \mathcal{L}_H(\mathit{SYSTEM} \parallel Z_c \parallel J_x)$$

Fix some $c \in C$ and $Z_{x_2} \in I_2$. By the definition of dominance, there is some $Z_{x_1} \in I_1$ such that $Z_{x_2} \sqsubseteq_{FD} Z_{x_1}$. Since *SYSTEM* meets $\mathit{CR}(I_1, C, H)$, there exist some $Z_c \sqsupseteq_{FD} \mathit{VOTER}(0, c)$ and $J_x \sqsupseteq_{FD} J$ such that

$$\mathcal{L}_H(\mathit{SYSTEM} \parallel Z_{x_1} \parallel J) \sqsubseteq_{FD} \mathcal{L}_H(\mathit{SYSTEM} \parallel Z_c \parallel J_x)$$

But this choice of Z_c and J_x will suffice to show that *SYSTEM* also meets $\mathit{CR}(I_2, C, H)$. For since $Z_{x_2} \sqsubseteq_{FD} Z_{x_1}$, the monotonicity of the CSP operators tells us that

$$\mathcal{L}_H(\mathit{SYSTEM} \parallel Z_{x_2} \parallel J) \sqsubseteq_{FD} \mathcal{L}_H(\mathit{SYSTEM} \parallel Z_{x_1} \parallel J)$$

and so by the transitivity of refinement we also have

$$\mathcal{L}_H(\mathit{SYSTEM} \parallel Z_{x_2} \parallel J) \sqsubseteq_{FD} \mathcal{L}_H(\mathit{SYSTEM} \parallel Z_c \parallel J_x)$$

■