A Formal Framework for Modelling Coercion Resistance and Receipt Freeness

James Heather and Steve Schneider

University of Surrey, Guildford, Surrey, GU2 7XH, UK

Abstract. Coercion resistance and receipt freeness are critical properties for any voting system. However, many different definitions of these properties have been proposed, some formal and some informal; 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, and strong enough to cover both randomization attacks and forced abstention attacks. 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, and the importance of understanding the relations between them.

1 Introduction

Much work has been published over the last couple of decades concerning secure voting protocols. Many proposals come with claims that they meet appropriate security guarantees; but the properties in question 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 [MN06,DKR09,DLL11]. 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, and to verify it against a formalization of the relevant property.

However, since the informal definitions vary considerably, 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 commonly discussed properties—coercion resistance and receipt freeness—and construct a framework that is rich enough to cope with a large variety of definitions. This has the advantage of allowing us to formalize many definitions and analyze a voting system to see which definitions it satisfies and which it does not. 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.

1.1 Characterisations

Characterisations of coercion resistance and receipt freeness are plentiful in the literature, but rarely do two definitions coincide. The following characterisations are examples from the literature. They are a mixture of coercion resistance and receipt freeness definitions; once we have seen the flavour of some of these definitions, then we will consider the differences. They apply to systems which voters interact with in order to cast votes, and which potential coercers may also observe and interact with.

Characterisation 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' [Oka97].*

Characterisation 2 (Benaloh/Tuinstra [RF]) *A voter should be unable to prove that a vote was cast in a particular way [BT94].*

Characterisation 3 (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 [DKR09]. (A weaker notion of receipt freeness is also provided.)*

The issue here is what can qualify as instruction. 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. First, 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. Secondly, 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 [Riv06]) then they might not.

Because receipt freeness properties are, on this understanding, a subclass of coercion resistance properties, we will focus on the larger problem of coercion resistance. Receipt freeness is discussed in more detail in an expanded and more technical version of this paper [HS12].

2 Modelling voting systems in CSP

CSP (Communicating Sequential Processes) provides a language for describing concurrent systems, and a theory for reasoning about them, in terms of events that they can perform. Events can be atomic, e.g., *start*, or they can be structured with several fields, e.g., $vote.i.p.v$. The language of processes includes:

- $a \rightarrow P$, which can perform a and then behave as P;
- $c?v \rightarrow P(v)$ which inputs value v over channel c and then behaves as $P(v)$;
- $c!v \rightarrow P$ which outputs v on channel c;
- $P \setminus A$ which hides the set of events in A, which are performed internally;
- *Stop* is the process that does nothing.
- $Chaos(H)$ is the process that can nondeterministically perform or refuse to perform any event from H at any time.
- $P \sqcap Q$ makes an internal (nondeterministic) choice between P and Q;
- $P \Box Q$ offers an external choice between P and Q;
- $P \parallel Q$ runs P and Q in parallel, synchronising on their common events.

The last three operators also have indexed forms. The language also includes recursive definitions $N = P$.

The theory provides a hierarchy of semantic models, including the Stable Failures model, which models a process as the set of traces (sequences of events) and subsequent sets of events that can be refused and the Failures/Divergences model which also includes information about divergent (infinite internal) behaviour. A process P is refined by another process Q, written $P \sqsubseteq_F Q$ and $P \sqsubseteq_{FD} Q$ for the respective models, if all observations of Q in that model are also observations of P. For further details, the reader is referred to [Ros98,Sch99].

Throughout this section, we shall assume that voting systems are modelled as follows. The system as a whole is modelled by a CSP 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, and 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; in the second case, 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 [DKR09].

The idea will be that Juan will cover Zara's tracks. 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 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 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. 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 4 (System model) *The system is modelled by a process $SYSTEM$, and the most general behaviour of a voter with ID i who chooses to vote for candidate c is modelled by $VOTER(i, c)$. 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*

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

These processes must meet the following conditions:

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

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

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

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 [CRS05], or unbounded revoting in a system like Helios [Adi08]. 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. Hence for the remainder of this paper we will use the stable failures semantic model.

Definition 5 (Coercer's control) *We use 'H' for the set of events invisible to the coercer. The only restriction 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 6 (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 7 *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:*

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

Definition 8 (Coercion resistance [*CR*]**)** *Suppose that we are given some system model SYSTEM (with associated voter model* $VOTER(i, c)$*).*

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

$$\begin{aligned} I &\subseteq \{P \mid P \sqsupseteq_F VOTER(0, C)\} \\ C &\subseteq CANDIDATES \\ H &\subseteq \Sigma \setminus \{open, close\} \\ mask &\subseteq (H \times H) \cup (\bar{H} \times \bar{H}) \end{aligned}$$

if, for all $c \in C$ *and* $Z_x \in I$*, there exist some* $Z_c \sqsupseteq_F VOTER(0, c)$ *and* $J_x \sqsupseteq_F VOTER(1, C)$ *such that*

$$\begin{aligned} &\mathcal{L}_H(mask(SYSTEM \parallel Z_x \parallel VOTER(1, C))) \\ &\qquad \sqsubseteq_F \quad \mathcal{L}_H(mask(SYSTEM \parallel Z_c \parallel J_x)) \end{aligned} \tag{1}$$

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 $I \subseteq \{P \mid P \sqsupseteq_F VOTER(0, C)\}$ 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 [Ros98]; it provides a mechanism for masking all of the events (in traces and in refusals) from the hidden set H. It is defined as $\mathcal{L}_H(P) = (P \parallel Chaos(H)) \setminus H$. 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. This is a stronger form of abstraction than simply hiding events, for which hidden events cannot be refused.

The purpose of the fourth parameter, *mask*, is to allow us to model semantic security of an encryption function. There will be times when we want to say that encryptions are essentially opaque to an observer: he cannot learn anything from seeing an encryption, including determining whether two encryptions represent the same value. The *mask* function is applied to events, and then lifted to whole processes; usually it will involve mapping all encryptions to a single value. The conditions on it state that it should act reasonably with respect to the events that are being entirely abstracted away: it will not move a whole event from hidden (in H) to visible (in $\Sigma \setminus H$) or from visible to hidden. For most of the models in this paper, we will use the identity function *id* as the mask, because there is no encryption to deal with; but for the Prêt à Voter model in Section 4, we will need to mask encryptions from the observer's view.

What Definition 8 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 9 (Coercion resistance $[CR^*]$**)** *The coercion resistance property* $CR^*(I, C, H, mask)$ *ihas the same definition as CR of Definition 8 except that*

it uses equality instead of refinement, replacing Line (1) with the following:

$$\exists\, J_c \sqsupseteq_F \mathit{VOTER}(1, c).$$
$$\mathcal{L}_H(\mathit{mask}(\mathit{SYSTEM} \parallel Z_x \parallel \mathit{VOTER}(1, C))) \sqsubseteq_F \mathcal{L}_H(\mathit{mask}(\mathit{SYSTEM} \parallel Z_c \parallel J_x))$$

In Definition 8, 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; in Definition 9, 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 9 is stronger than Definition 8 since equality implies refinement. For most voting systems there will be no difference; but we will see in Section 4.1 that this is not always the case. Hence we illustrate the difference between approaches based on CR and those based on CR^*.

The line we will adopt here is to use Definition 8 for the bulk of our work, to illustrate how the definition can be applied. Similar results hold for Definition 9.

3 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 1.1. 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 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^*.

3.1 Formal definitions

For convenience, we will attach a superscript of 'abs' when the definition includes the special abstention candidate. The definitions below are given in their undecorated form; 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 4.

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 10 (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.

Definition 11 (Okamoto) *Characterisation 1 is encapsulated formally by using the following set within the definition of CR or CR∗:*

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

The coercer may specify a candidate to vote for, but may not specify how the voter is to cast it. This turns out to be equivalent to I_{RFGEN}.

Definition 12 (Benaloh/Tuinstra) *We here give the formal definition of coercion resistance for Characterisation 2. 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 encapsulated by using the following set within the definition of CR or CR*:*

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

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 13 (Delaune/Kremer/Ryan) *Characterisation 3 says that a system is coercion resistant if the coercer cannot tell whether a coerced voter has behaved as instructed or voted differently. 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 of the set I within our framework is then the same as that for the Benaloh/Tuinstra definition: the coercer can choose any candidate, then specify 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 always be the appropriate definition corresponding to theirs.*

Definition 14 (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: $I^{\mathsf{abs}} = \{\, VOTER(0, \mathsf{abs})\}$.*

Definition 15 (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.*

$$I_{MAX} = \{P \mid P \sqsupseteq_F VOTER(0, C)\} \quad \textit{where } \mathsf{abs} \notin C$$
$$I^{\mathsf{abs}}_{MAX} = \{P \mid P \sqsupseteq_F VOTER(0, C)\} \quad \textit{where } \mathsf{abs} \in C$$

The possibility of randomisation attacks is dependent on the particular system under consideration, and there is not a generic characterisation. We see an example of a randomisation set I_{RND} in the next section, under Proposition 20.

3.2 Relationships between definitions

Some of the associated CR and $CR*$ definitions are stronger than others. We now state some results that allow us to formalize relations between notions of coercion resistance. Proofs of the results given in this paper can be found in [HS12].

Definition 16 (Dominance) *Suppose that I_1 and I_2 are sets of processes. We say that I_1* dominates *I_2 if $\forall\, P_2 \in I_2.\, \exists\, P_1 \in I_1 . P_2 \sqsubseteq_F P_1$.*

Theorem 17 (CR and dominance) *Suppose that I_1 dominates I_2, and SYSTEM meets $CR(I_1, C, H, mask)$. Then it also meets $CR(I_2, C, H, mask)$.*

Corollary 18 (CR and subset) *Suppose that $I_2 \subseteq I_1$, and SYSTEM meets $CR(I_1, C, H, mask)$. Then it also meets $CR(I_2, C, H, mask)$.*

These results allow us to give a hierarchy of definitions, whose relationships are shown in Figure 1

4 Example: Simplified Prêt à Voter

Figure 2 shows the CSP for a simplified model of Prêt à Voter running a referendum. Voters receive a value $b \in \{0, 1\}$ on channel *ballot*, which indicates the ordering of the boxes on the ballot form: 0 or 1 means that the top box represents 'yes' or 'no' respectively. They also receive a pair of encryptions, the first (second) of which will decrypt to the value represented by the top (bottom) box. They then submit an ID from the finite set *IDS* of all voter IDs, and the encryption associated with the box they want to choose; the system returns this encryption to them, and then stores the encrypted value.

When voting closes, the set of votes is passed through each of the K mix servers in turn, which each re-encrypt them all. The votes are then decrypted and the totals announced.

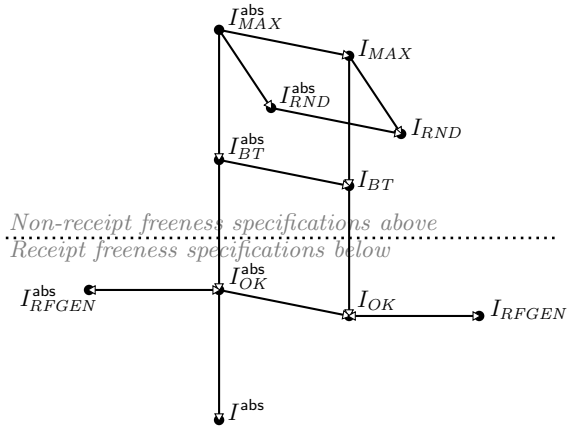

Fig. 1. Hierarchy of definitions of coercion resistance

Here and throughout, '$\bar{v}$' represents $1 \oplus v$, where '$\oplus$' is bitwise exclusive-or. (The special candidate abs is treated as invariant under this operation.) The ballots and re-encryptions are produced by *ENCSVR*, which models the assumption that no two encryptions ever have the same randomness. It is initialized with an infinite sequence *RANDS* of distinct random numbers, and it uses these to generate new ballots and re-encryptions of existing ballots.

The voter process is finitary (Assumption 4) because it is non-recursive. The system is finite because on each 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 abstentions into account.

For what follows, we define the function *mask* to model the semantic security of the encryption: it abstracts the system so that all encryptions appear as the value '$\perp$'. This prevents agents from 'reading' inside the encryptions:

$$mask(ballot.i.b.E) = ballot.i.b.\langle \perp \mid e \in E \rangle \quad mask(mix.n.B) = mix.n.\{\perp \mid b \in B\}$$
$$mask(vote.i.p.v) = vote.i.p.\perp \qquad mask(write.v) = write.\perp$$

Proposition 19 (Okamoto and PaV, no abs) *The set of candidates under consideration, when abstentions are not taken into account, is $C_2 = \{0, 1\}$.*

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 masked 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.

$$
\begin{aligned}
BOOTH &= open \rightarrow WAITING(\emptyset, \emptyset) \\
WAITING(VTD, BOX) &= arrive?id : IDS \setminus VTD \rightarrow VOTING(VTD, BOX, id) \\
&\quad \Box\ close \rightarrow mix.1!BOX \rightarrow Stop \\
VOTING(VTD, BOX, id) &= newBallot \rightarrow genBallot?b?encs \rightarrow ballot.id!b!encs \\
&\quad \rightarrow \Box_{v \in \{0,1\}} (vote.id.v.encs[v] \rightarrow write!encs[v] \rightarrow \\
&\qquad\qquad WAITING(VTD \cup \{id\}, BOX \cup \{encs[v]\}))
\end{aligned}
$$

$$
\begin{aligned}
MIX(n) &= close \rightarrow mix.n?BOX \rightarrow REMIX(n, BOX, \emptyset) \\
REMIX(n, \emptyset, NEW) &= mix.(n+1)!NEW \rightarrow Stop \\
REMIX(n, OLD, NEW) &= \sqcap_{v \in OLD} reenc.n!v \rightarrow genReenc.n?v' \rightarrow \\
&\qquad\qquad REMIX(n, OLD \setminus \{v\}, NEW \cup \{v'\})
\end{aligned}
$$

$$
DEC = mix.(K+1)?BOX \rightarrow announce!cnt(BOX, 0).cnt(BOX, 1) \rightarrow Stop
$$

$$
\begin{aligned}
SYSTEM = (BOOTH \parallel \left(\|_{1 \leqslant i \leqslant K} MIX(i) \right) \parallel ENCSVR(RAND) \parallel DEC) \\
\setminus \{|newBallot, genBallot, reenc, genReenc|\}
\end{aligned}
$$

$$
\begin{aligned}
ENCSVR(R) &= newBallot \rightarrow \sqcap_{b \in \{0,1\}} genBallot!b!\langle Enc.b.R[0], Enc.(1-b).R[1] \rangle \rightarrow ENCSVR(R[2..]) \\
&\quad \Box\ reenc?n?Enc.b.r \rightarrow genReenc!n!Enc.b.R[0] \rightarrow ENCSVR(R[1..]) \\
VOTER(i, c) &= open \rightarrow \text{if } (c \neq \mathsf{abs}) \text{ then} \\
&\qquad arrive!i \rightarrow ballot.i?b?xs \rightarrow vote.i.c \oplus b.xs[c \oplus b] \rightarrow close \rightarrow Stop \\
&\quad \text{else } close \rightarrow Stop \\
cnt(BOX, b) &= \#\{r \mid Enc.b.r \in BOX\}
\end{aligned}
$$

Fig. 2. A simplified model of Prêt à Voter: defining the system and voter behaviour

The simplified Prêt à Voter model meets $CR(I_{OK}, C_2, H_{PUB}, mask)$.

Proposition 20 (Randomization attacks and PaV, no abs**)** *To mount a randomization attack, the coercer specifies a particular box to be ticked (for instance, the top box). The coercer cannot know whether this box represents a 'yes' or a 'no' vote. Such an attack is represented in our model by setting*

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

We consider candidates in $C_2 = \{0, 1\}$*, and* $H_{PUB} = \{|ballot|\}$*. The coercer can see which box Zara ticks, but not which candidate it represents.*

Our simplified model of Prêt à Voter does not meet $CR(I_{RND}, C_2, H_{PUB}, mask)$.

Corollary 21 (I_{MAX} **and PaV, no** abs**)** *It is an immediate corollary of Proposition 20 and Corollary 18 that our simplified Prêt à Voter does not meet* $CR(I_{MAX}, C_2, H_{PUB}, mask)$.

Any set of coercer instructions must be a subset of I_{MAX}, so Corollary 18 tells us that if Prêt à Voter met $CR(I_{MAX}, C_2, H_{PUB}, mask)$ then it would meet $CR(I, C_2, H_{PUB}, mask)$ for any I. But Proposition 20 shows that it does not meet $CR(I_{RND}, C_2, H_{PUB}, mask)$; therefore, it cannot meet $CR(I_{MAX}, C_2, H_{PUB}, mask)$.

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. First, Zara may now want to abstain; coercion resistance will imply that she is able to abstain if she wishes, without the coercer knowing that she has not complied. If the coercer can force Zara not to abstain, then we have a *forced participation* attack. Secondly, 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 22 (Okamoto and PaV, C_2^{abs}) *The Okamoto definition, with* abs *included, is modelled by*

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

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.*

Our simplified Prêt à Voter model does not meet $CR(I_{OK}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{PUB}, mask)$. If the coercer can see anything that includes the voter's ID, then there is no hope of resistance to forced abstention attacks.

Corollary 23 (I_{MAX}^{abs} and PaV, C_2^{abs}) *The strongest definition, with* abs *included, is modelled by $I_{MAX}^{\mathsf{abs}} = \{P \mid P \sqsupseteq_{FD} VOTER(0, c) \mid c \in C_2^{\mathsf{abs}}\}$*

Our model does not meet $CR(I_{MAX}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{PUB}, mask)$.

Finally we ask what happens if we change the level of abstraction, so that the coercer can see fewer events. We will allow the coercer to see votes being posted up (on the *write* channel), but not arrivals or vote casting. We will set $H_{SEC} = \{|arrive, ballot, vote|\}$.

Proposition 24 (I_{MAX} and PaV, C_2^{abs}, H_{SEC}) *Our simplified Prêt à Voter model meets $CR(I_{MAX}, C_2^{\mathsf{abs}}, H_{SEC}, mask)$. 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.*

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 1.

Definition	Abs?	Invisible	Formalism	Met by PaV?
Okamoto	No	$\{\|ballot\|\}$	$CR(I_{OK}, C_2, H_{PUB}, mask)$	Yes
Randomization	No	$\{\|ballot\|\}$	$CR(I_{RND}, C_2, H_{PUB}, mask)$	No
Strongest	No	$\{\|ballot\|\}$	$CR(I_{MAX}, C_2, H_{PUB}, mask)$	No
Okamoto / forced abs	Yes	$\{\|ballot\|\}$	$CR(I_{OK}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{PUB}, mask)$	No
Strongest	Yes	$\{\|ballot\|\}$	$CR(I_{MAX}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{PUB}, mask)$	No
Strongest	Yes	$\{\|arrive, ballot, vote\|\}$	$CR(I_{MAX}^{\mathsf{abs}}, C_2^{\mathsf{abs}}, H_{SEC}, mask)$	Yes

Table 1. Summary of results for simplified Prêt à Voter model

4.1 Further Examples

Two further toy examples illustrate the differences between 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^*. In each case, we give here the informal definitions and state the properties the systems meet; the CSP models can be found in [HS12].

Two-receipt This system gives voters a receipt containing two names (in arbitrary order): the name 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.

Two-receipt meets the property $CR(I_{OK}, \{c_1, c_2, c_3\}, \{|vote, dummy|\}, id)$. A voter instructed to vote for c' can vote for c in a way consistent with a vote for c'. Conversely, Two-receipt does not meet the Benaloh and Tuinstra characterisation as captured by the property $CR(I_{BT}, \{c_1, c_2, c_3\}, \{|vote, dummy|\}, id)$. This is 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.

Opt-receipt The following example is attributed to Ron Rivest. 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 consistently with any instructions a coercer might give them, by declining any receipt, and claiming that the system did not offer one.

The voter has a strategy for voting without production of a receipt, and so Opt-receipt meets $CR(I_{MAX}, C, H_{OPT}, id)$, where $H_{OPT} = \{|vote, noreceipt, offerreceipt, accept, reject|\}$. However, it does not meet $CR^*(I_{MAX}, C, H_{OPT}, id)$. 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.

5 Discussion

As commented in Section 1, there are a variety of definitions in the literature to receipt-freeness and coercion-resistance, and a range of approaches to analysing proposed voting protocols and systems. They all hinge on the required inability of a coercer to tell whether the coerced voter has followed instructions or not.

The game-based approach typically applied to cryptographic schemes has been applied with respect to coercion-resistance in [JCJ05,GGR09,KTV10]. In this approach, coercion-resistance is captured in terms of a game with a specific goal, e.g. [GGR09] considers Indistinguishability of Encoded Votes. The nature of the goal and the coercer's possible instructions characterises whether abstention and randomisation attacks are included, and so the hierarchy of Figure 1 also applies to the range of possibilities expressible using the game-based approach.

Coercion resistance has also been characterised in the Universal Composability (UC) framework, for example in [MN06], [UMQ10], [dMPQ07]. This approach uses an idealised system in which voters choose whether or not to obey the coercer, and then defines a coercion-resistant system to be one in which an adversary cannot enable a distinguisher to tell the difference between the real system and the idealised system. Though in a different setting, this gives the same sense of coercion-resistance as Definition 3. The hierarchy of definitions of Figure 1 for the UC setting corresponds to what the coercer can require of the voter in the idealised system. Abstention attacks fall naturally within this setting, but randomisation attacks will perhaps be more difficult to characterise.

Others take a more symbolic approach. Okamoto's original formulation [Oka97] was epistemic. More recently the epistemic approach of [KT09] requires that for any instructions provided by the coercer, there is a counter-strategy for the voter to achieve their own goal, where the coercer cannot tell whether or not his instructions were followed. The hierarchy of definitions arises naturally with this approach, as the possible instructions and observations of the coercer vary. A quantitative approach based on knowledge reasoning is given in [JMP09], which gives a measure of voter privacy.

The process algebraic approaches of [DKR09,BHM08] and this paper are also symbolic. In these approaches an observational equivalence is used for indistinguishability, and coercion-resistance is captured as the equivalence of two processes, one where the voter complies and one where he does not. The models in [DKR09] provide a general framework to include the weaker properties of receipt-freeness and privacy, but unlike our approach they do not handle abstention or randomisation attacks since they are characterised in terms of a coercer selecting a particular candidate. The extended framework of [BHM08] does explicitly handle forced abstention attacks, but not randomisation attacks.

Our approach is most closely related to the epistemic characterisation in [KT09], but ours is cast in a process algebraic setting. This allows a higher level description of a voting system design in CSP. Further, our emphasis is on the hierarchy of definitions rather than the proposal of any specific one.

Acknowledgements We are grateful to the reviewers for their careful reviewing and their valuable comments and suggestions. Thanks are also due to Chris Culnane, Murat Moran, Sriram Srinivasan and Zhe (Joson) Xia. 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

[Adi08] Ben Adida. Helios: Web-based open-audit voting. In Paul C. van Oorschot, editor, *USENIX Security Symposium*, pages 335–348, 2008.

[BHM08] Michael Backes, Catalin Hritcu, and Matteo Maffei. Automated verification of remote electronic voting protocols in the applied pi-calculus. In *CSF*, pages 195–209. IEEE Computer Society, 2008.

[BT94] Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC*, pages 544–553, 1994.

[CRS05] D. Chaum, P. Y. A. Ryan, and S. Schneider. A practical voter-verifiable election scheme. In *ESORICS*, volume 3679 of *LNCS*. Springer, 2005.

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

[DLL11] Jannik Dreier, Pascal Lafourcade, and Yassine Lakhnech. A formal taxonomy of privacy in voting protocols. Technical Report TR-2011-10, Verimag, 2011.

[dMPQ07] Olivier de Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Simulation-based analysis of e2e voting systems. In *VOTE-ID*, volume 4896 of *LNCS*, pages 137–149. Springer, 2007.

[GGR09] Ryan W. Gardner, Sujata Garera, and Aviel D. Rubin. Coercion resistant end-to-end voting. In *Financial Cryptography*, volume 5628 of *LNCS*, 2009.

[HS12] J. Heather and S.A. Schneider. A formal framework for modelling coercion resistance and receipt freeness. Technical report, University of Surrey, 2012.

[JCJ05] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *WPES*, pages 61–70, 2005.

[JMP09] Hugo L. Jonker, Sjouke Mauw, and Jun Pang. A formal framework for quantifying voter-controlled privacy. *J. Algorithms*, 64(2-3):89–105, 2009.

[KT09] Ralf Küsters and Tomasz Truderung. An epistemic approach to coercion-resistance for electronic voting protocols. In *IEEE Symposium on Security and Privacy*, pages 251–266. IEEE Computer Society, 2009.

[KTV10] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. A game-based definition of coercion-resistance and its applications. In *CSF*, pages 122–136. IEEE Computer Society, 2010.

[MN06] Tal Moran and Moni Naor. Receipt-free universally-verifiable voting with everlasting privacy. In *CRYPTO*, volume 4117 of *LNCS*. Springer, 2006.

[Oka97] Tatsuaki Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Security Protocols Workshop*, LNCS. Springer, 1997.

[Riv06] Ronald L. Rivest. The ThreeBallot Voting System. `http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf`, 2006.

[Ros98] A.W. Roscoe. *Theory and Practice of Concurrency*. Prentice-Hall, 1998.

[Sch99] S. Schneider. *Concurrent and Real-time Systems*. Wiley, 1999.

[UMQ10] Dominique Unruh and Jörn Müller-Quade. Universally composable incoercibility. In *CRYPTO*, LNCS. Springer, 2010.