Automated Anonymity Verification of the ThreeBallot and VAV Voting Systems

Murat Moran[1], James Heather[2], and Steve Schneider[3]

[1] Giresun University, Turkey, muratmoran@gmail.com
[2] Chiastic Security, UK, drfudgeboy@gmail.com
[3] University of Surrey, UK, s.schneider@surrey.ac.uk

Abstract. In recent years, a large number of secure voting protocols have been proposed in the literature. Often these protocols contain flaws, but because they are complex protocols, rigorous formal analysis has proven hard to come by.
Rivest's ThreeBallot and Vote/Anti-Vote/Vote (VAV) voting systems are important because they aim to provide security (voter anonymity and voter verifiability) without requiring cryptography. In this paper, we construct CSP models of ThreeBallot and VAV, and use them to produce the first automated formal analysis of their anonymity properties.
Along the way, we discover that one of the crucial assumptions under which ThreeBallot and VAV (and many other voting systems) operate—the short ballot assumption—is highly ambiguous in the literature. We give various plausible precise interpretations, and discover that in each case, the interpretation either is unrealistically strong, or else fails to ensure anonymity. We give a revised version of the short ballot assumption for ThreeBallot and VAV that is realistic but still provides a guarantee of anonymity.

Keywords: Formal Methods, Voting Systems, FDR2, CSP, Anonymity, Automatic Verification, ThreeBallot, VAV

1 Introduction

Recent years have seen a large number of end-to-end voting systems proposed in the literature [1, 2, 3, 4, 5]. Typically these systems aim to provide a proof of correctness of the election tally, but also some guarantee of privacy for the voter; and cryptography is usually employed to achieve these goals. Rivest's ThreeBallot and VAV voting systems [6] is particularly interesting because it uses no cryptography, but nevertheless still aims to provide anonymity, integrity of the election, verifiability and incoercibility.[4]

One of the most critical properties of voting systems is anonymity, which essentially requires that the link between voters and votes be broken. Anonymity is important for voter privacy as well as being essential for preventing coercion and vote buying. This paper considers the anonymity property as it relates to the ThreeBallot and VAV voting systems [6].

ThreeBallot and VAV rely heavily on the *short ballot assumption* (SBA) to assist in providing its anonymity guarantee. Roughly speaking, this assumption states that the information content of a ballot should be low. However, the phrasing of this assumption in the description of ThreeBallot is vague, and open to a number of radically different interpretations. We consider the various possibilities here. Some turn out to be unrealistically strong; some seem to be too weak to guarantee anonymity. In the process, we construct formal models of ThreeBallot and VAV in Communicating Sequential Processes (CSP) [7], and use the Failures-Divergences Refinement (FDR2) model checker [8] to produce an automated analysis of the model. Some other voting systems have been at least partially verified automatically against privacy-related properties (for example, Civitas [3] in [9] with hand-proofs, FOO [2] in [10] with a compiler, and Prêt à Voter [11] in [12]); but the ThreeBallot and VAV voting systems have not yet been subjected to automated formal verification.

The paper is constructed as follows. In the remainder of this section, we give an outline of ThreeBallot, and discuss related work. In Sect. 2, we model ThreeBallot as a parallel composition of agents: voters, an authority, and a bulletin board. Then, using an anonymity definition given in [12], in Sect. 3.1 we analyse our model against an adversary who can observe all public channels. Initially, our model drops the SBA entirely,

[4] A previous version of this paper appeared in the Proceedings of the 10th International Conference on Integrated Formal Methods, Springer LNCS 7940. The final publication is available at `http://link.springer.com/chapter/10.1007%2F978-3-642-38613-8_7`

and we discover that FDR leads us to several attacks on vote anonymity. Section 3.2 then discusses the SBA in its various guises, and shows that in each case the assumption is either too strong to be realistic or too weak to be secure; we then propose a different short ballot assumption that is both reasonable and demonstrably strong enough to provide anonymity. In the Sect. 3.3 we analyse the other versions of ThreeBallot, and demonstrate that with the modifications, ThreeBallot provides guaranteed anonymity. Section 4 evaluates the VAV voting system by first describing how it works, then focuses on the modelling of VAV in CSP. Moreover, results from the automated analysis of VAV are given in Sect. 4.2 and the SBA for VAV is discussed in Sect. 4.3. Finally, Sect. 5 concludes this paper with a summary of findings and present limitations.

1.1 Voting with ThreeBallot

In this section, we briefly introduce the original ThreeBallot voting system and the short ballot assumption given by Rivest and Smith [6].

Voting in ThreeBallot proceeds as follows. Initially, the (authenticated) voter receives a multi-ballot form from a pollworker, which consists of three mini-ballot forms (see Fig. 1). The mini-ballots are all identical except for the IDs or serial numbers, located at the bottom of the mini-ballots. These serial numbers are all unique, and are not meaningful. In particular, there is no way of determining what mini-ballot serial numbers go together to make up a multi-ballot.

The voter fills two bubbles in total for the chosen candidate, and only one bubble for each other candidate. The completed multi-ballot is inserted into a checker, which confirms that it has been correctly completed.

Finally, the voter chooses one of the mini-ballots, and receives a duplicate of that mini-ballot as her receipt. She then separates the three mini-ballots, and casts them all individually into a ballot box.

After the election, all mini-ballots are published on a web bulletin board, along with a list of everyone who voted. The voter may then verify that the mini-ballot for which she has a receipt appears unaltered on the bulletin board (BB); if it does not, she can produce the receipt as evidence of foul play.

Alice	●	Alice	●	Alice	○
Bob	○	Bob	●	Bob	○
Chris	○	Chris	○	Chris	●
David	●	David	○	David	○
	56248		04578		31489

Fig. 1. A ThreeBallot multi-ballot, filled in as a vote for Alice

The number of votes for each candidate is counted as usual. However, as each voter fills in exactly two bubbles for the chosen candidate and one bubble for the other candidates, the number of voters is subtracted from each candidate's final tally to find the correct number of votes for each candidate. Since all the mini-ballots are posted on the bulletin board, the final tally can be verified by anyone.

ThreeBallot is claimed in [6] to be secure under the short ballot assumption (SBA). Rivest and Smith define the SBA as the assumption that

> the ballot is short—there are many more voters in an election than ways to fill out an individual ballot [...] It is reasonable to assume under the SBA that each possible ballot is likely to be cast by several voters.

The ambiguities arise from the terms "possible ballots" (mini-ballots or multi-ballots?) and "several voters" (how many?).

Looking elsewhere for clarification bears little fruit. According to Cichoń, Kutylowski, and Weglorz [13] the SBA assumes that "the list of candidates on a ballot is short enough in order to guarantee security"; we read in [14] that "the length of the ballots must be kept small (possibly by splitting them into several parts)".

Because ThreeBallot is claimed to guarantee voter anonymity under the SBA, analysis of ThreeBallot is not possible without a clear and unambiguous reading of the assumption. We give here three possible interpretations; we will analyse ThreeBallot under each of these readings in Sect. 3.2.

In each case, the intention is that the assumption will be guaranteed probabilistically; that is, that the number of voters, candidates, etc., will be sufficient to ensure that the assumption is broken with only negligible probability. In what follows, serial numbers will be ignored; that is, two mini-ballots will be considered the same if they contain the same marks apart from the serial numbers.

Assumption 1 (SBA-multi) *Every possible multi-ballot will be cast at least once.*

The formulation of the SBA given in Assumption 1 requires that every possible way of completing a multi-ballot should be adopted by at least one voter. For small numbers of candidates, this is not implausible. For even moderate numbers, though, the assumption quickly becomes hard to stomach.

Note that once one has chosen a candidate, there are then exactly three ways of completing each row: for the chosen candidate's row, one must choose a bubble to leave empty, and for each other row, one must choose a bubble to fill. There are thus $c \cdot 3^c$ distinct multi-ballots, where c is the number of candidates standing in the election.

It is not feasible to calculate the number of voters required to make this reasonable, because it depends on the probability distribution of multi-ballots: voters do not cast multi-ballots randomly (one hopes). A full calculation would require a realistic model of how voters cast their ballots. However, the best case scenario is when voters cast their multi-ballots randomly; so by assuming a uniform distribution, we can determine a lower bound on the number of voters required.

With a uniform distribution, the expected number of voters needed to cover all possible multi-ballots is $n \cdot \sum_{i=1}^{n} \frac{1}{i}$ where $n = c \cdot 3^c$, the number of possible multi-ballots. For five candidates, this comes out at 9331 voters; for ten candidates, we need 8.1 million voters; for fifteen candidates, the number exceeds 4 billion.

For n possible multi-ballots, and a uniform distribution, we can calculate the number of voters required to ensure that the probability of covering every multi-ballot at least once exceeds a given threshold. Since the security of ThreeBallot relies on the SBA, we would need confidence that (the correct interpretation of) the SBA is satisfied; we can, therefore, for a given probability level, ask how many voters are required to give this level of confidence that the SBA will be satisfied.

For n multi-ballots, and v voters, the probability that the v voters will cover all of the n possibilities is

$$1 - \sum_{j=1}^{n-1} (-1)^{j+1} \binom{n}{j} \left(\frac{n-j}{n} \right)^v$$

This sum is difficult to calculate precisely but easy to calculate approximately because the first few terms dominate for large v.

For five candidates, to reach 95% probability of full coverage, we need around 12,250 voters. Six candidates need around 50,000 voters; by the time we reach ten candidates, 9.6 million voters are required to give 95% confidence that every multi-ballot turns up at least once. Note that these figures are rather conservative lower bounds: the distribution will not in fact be uniform, which will lower the probability; and in any case 95% confidence is perhaps insufficient for a critical security assumption.

These numbers are very high, and we consider them to be unrealistic. This version of the short ballot assumption is suitable only for a very small number of candidates or extremely large numbers of voters; it will not be considered further in this paper.

Assumption 2 (SBA-mini) *Every possible mini-ballot will be cast at least once.*

Under Assumption 2, we require only that each mini-ballot, rather than each multi-ballot, be cast. Clearly this is more likely to be satisfied than Assumption 1. For c candidates, there are only 2^c distinct mini-ballots, against $c \cdot 3^c$ distinct multi-ballots. For ten candidates, we therefore need coverage of only 1024 mini-ballots, rather than nearly 600,000 multi-ballots.

We will show later that this interpretation of the SBA is insufficient to prevent attacks on ThreeBallot. Since it is not a worthwhile formulation of the assumption, we need not calculate the likelihood that it will be satisfied.

Assumption 3 (SBA-mini-t) *Every possible mini-ballot will be cast at least t times (for some suitably chosen t).*

A slightly stronger interpretation in Assumption 3 requires each mini-ballot to turn up at least a certain number of times. This, of course, requires more voters than Assumption 2.

However, we will show later that this formulation is also insecure, regardless of the value of t.

1.2 Related Work

The ThreeBallot voting system has been subjected to analysis of one sort or another many times since its publication [13, 14, 15, 16, 17, 18, 19, 20, 21]. Perhaps the earliest analysis was conducted by Strauss [15, 16], who established the success probabilities of attacks for various numbers of candidates and voters with multiple races. Various attacks against the system, and in particular, reconstruction and pattern request attacks, were considered. The experiments were coded in Python, and modelled elections with a number of races on a single multi-ballot form. Clark *et al.* [17] also investigated ThreeBallot, and pointed out that the multi-ballot reveals information that can compromise voter privacy. A simulation-based analysis of the system was made by de Marneffe *et al.* [14] using the universally composable security framework [22]. Additionally, a modified system protocol in which a voter chooses her receipt before expressing her preference was proposed in [14]. This protocol was shown to guarantee election fairness, at the cost of some noise in the final tally, with the SBA assumption, and an additional assumption that most of the receipts are not known to the adversary. One drawback, however, is that the voter cannot express her preference on the mini-ballot that she has chosen as her receipt, which makes voting more complicated. Statistical results about the relation between the number of candidates in an election and the privacy level of the system were provided by Cichoń *et al.* [13] as well as a critique on the effectiveness of Strauss' attacks. Cichoń *et al.* claim that it is impossible to reconstruct voters' preferences in a single election run with two candidates with a 'reasonable number of voters'. However, their definition of anonymity used in [13] is much different from ours given in [12]. Considering that an individual mini-ballot can be used to construct two different multi-ballots cast for the same candidate, their definition seems necessary, but not sufficient. Hence, the observer would notice that one of the voters is not able to vote for that candidate as there is not enough mini-ballot that can be used to construct two valid multi-ballots for that candidate.

A more theoretical work was carried out by Henry *et al.* [20], who focused on a two-candidates race, and determined secure ballot sizes against reconstruction and pattern requesting attacks. Finally, Küsters *et al.* [21] computationally analysed the level of privacy offered by the ThreeBallot voting system and the proposed system by de Marneffe *et al.* [14], and concluded that the latter provides better privacy than the original.

2 Modelling the ThreeBallot Voting System

In this section, the CSP model of ThreeBallot [5] is given by first defining data-types, sets and the functions, and then describing each process individually. Subsequently, each is then run in parallel so as to reflect the behaviour of the voting system.

In this modelling approach, the multi-ballot of the ThreeBallot voting system is treated as a board with coordinates. Here, a co-ordinate (i, j) defines a bubble on a mini-ballot, which is to be filled in — although the

[5] The CSP model of ThreeBallot voting system, from which the experimental results given in this paper were produced, can be downloaded from the first author's personal webpage `http://muratmoran.wordpress.com/publications/` under the CSP codes title. It is also available on the departmental webpage `http://epubs.surrey.ac.uk/id/eprint/804928`.

bubbles at the end of each mini-ballot are allocated for serial numbers, the separation between a bubble for a mark and the bubble for a serial number is ensured in the process definitions. Thus, a multi-ballot consists of three columns each representing: a mini-ballot; as many rows as the number of candidates racing in the election; and a single row at the end of the ballot allocated for serial numbers. Hence, the size of the board is determined by these parameters: the number of voters and the number of candidates. These parameters define the sets of voters, candidates and serial numbers (there are three times as many serial numbers as there are voters). Conventionally, the data-types for voters, candidates and serial numbers are denoted v, c and s, respectively. Note that the candidate order is predetermined numerically, i.e., the order of the candidate is fixed as c_1,c_2, $\ldots$, c_n for all ballots.

In order to return a specific part of the board in a process description, several functions are used in the model. In more detail, the function $\mathsf{Row}(i)$ returns the ith row of a multi-ballot and $\mathsf{Col}(j)$ is the set of bubbles on the jth column (Fig. 2). Likewise, some other functions are used to return the neighbouring bubbles of a given coordinate, such as, the function $\mathsf{nhdAll}(i, j)$, which returns all the neighbours of (i, j) in the current multi-ballot coordinates (Fig. 3). Similarly, $\mathsf{adjR}(i, j)$ returns the coordinates adjacent to (i, j) in the same row (Fig. 4), and $\mathsf{adjC}(i, j)$ returns the coordinates adjacent to (i, j) in the same column (Fig. 5).

2.1 Modelling Assumptions

The assumptions made in the modelling of ThreeBallot are as the following. There is a limited number of voters, all of which follow the protocol steps honestly and choose the candidates to vote for before the registration phase. Additionally, the way that the voter fills in her ballot form is modelled in a way that is efficient in terms of the required state space for the automated analysis, while allowing the voter to cast any possible ballot form marked for the chosen candidate. Therefore, this modelling assumption does not impact on the analysis, but facilitates the mechanised

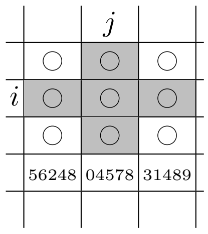

Fig. 2. Bubbles returned by Row(i) and Col(j)

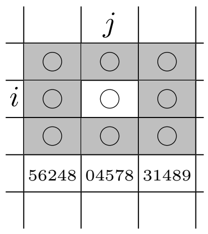

Fig. 3. Bubbles returned by nhdAll(i, j)

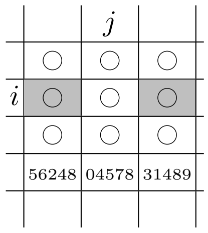

Fig. 4. Bubbles returned by adjR(i, j)

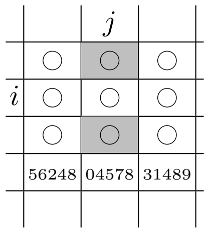

Fig. 5. Bubbles returned by adjC(i, j)

verification (more details are given when describing the voter process in the following subsection).

In the original ThreeBallot system there exists a checker machine in the booth, which confirms that the multi-ballot inserted by the voter has been correctly completed. In modelling of ThreeBallot, this behaviour is already modelled in the voter process — the voter never makes a mistake when completing a multi-ballot, which is ensured by process synchronisations, thus all cast multi-ballots are valid. This also means that we do not model incorrect voter behaviours either by mistake or intentionally in the voting system models analysed in this paper.

2.2 Honest Participants

In this section, we define how the ThreeBallot voting system model works, and explain what information is carried on each channel. The overall system model is a parallel composition of the processes detailed below. Figure 6 illustrates the network for the ThreeBallot CSP model.

Voter Process. The voter chooses the candidate that she wants to vote for before the election. She then authenticates herself to the election au-

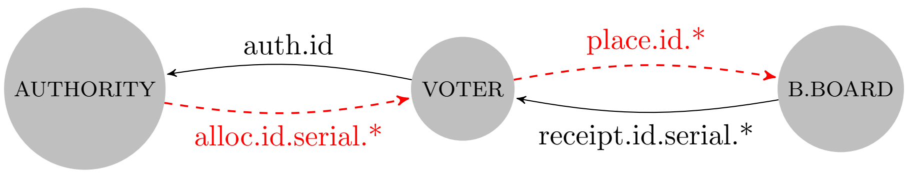

Fig. 6. ThreeBallot CSP Model Communication Channels (($\dashrightarrow$)private channel)

thority, and collects her multi-ballot on the *alloc* channel. In the booth, she fills out two bubbles for the chosen candidate and one for the other candidates with the *place* events. Afterwards, she gets her receipt by choosing one of the mini-ballots allocated to her on the channel *receipt*, and leaves the booth before the election is closed.

The *Voter*() process performs *place* events in an efficient way, such that although it allows any legitimate completed multi-ballot, it reduces the state space by constraining the order in which bubbles are filled. That is, first a bubble from the first or second column is chosen for the candidate the voter wants to vote for (the set F_1 under the second non-deterministic choice in the process definition below determines what bubbles the voter can use first, where $F_1 = \mathsf{Row}(x-1) \setminus \mathsf{Col}(2)$) [6] and x is the chosen candidate id, such as 2 for the candidate c_2, thus, the value $x - 1$ determines the row of the chosen candidate. Following this, the voter chooses another bubble for her chosen candidate in the same row, but in a left to right fashion, e.g., first marking the bubble $(0, 1)$, and then $(0, 2)$, thus eliminating the option of first marking $(0, 2)$ and then $(0, 1)$. This is illustrated by the voter choosing a bubble from the set F_2, where $F_2 = \mathsf{adjR}(i, j)$. Afterwards, the process performs a one *place* event in a top to bottom manner for the other candidates — again the way that the voters fill in the bubbles is restricted in order to reduce the state space required for the analysis. All empty bubbles that can be filled in by the voter in the next phase are transferred to the set *Rest*, where $Rest = \mathsf{nhdAll}(i, j) \setminus (\mathsf{Row}(i) \cup \mathsf{Row}(n))$ and n is the number of candidates, also corresponds to the serial numbers row on a ballot form. Once the voter has finished voting for the chosen candidate, she marks one

[6] We denote the set subtraction as $\setminus$ in order to distinguish it from the hiding operator "\" in CSP.

bubble for the other candidates, and the row belonging to the particular one is discarded from the set (modelled as $aset \setminus \mathsf{Row}(k)$ in the last line of the process definition). Finally, the voter determines one of the serial numbers as her receipt. That is, the mini-ballot with that serial number and the marked bubbles form her receipt.

$$
\begin{aligned}
&\mathit{Voter}(v) \mathrel{\widehat{=}} \\
&\quad \bigsqcap_{c \in \mathcal{C}} \left(\begin{array}{l}
\mathit{choose}.v.c \to \mathit{openElection} \to \mathit{auth}.v \to \\
\mathit{alloc}.v?s_1?(i_1, j_1) \to \mathit{alloc}.v?s_2?(i_2, j_2) \to \mathit{alloc}.v?s_3?(i_3, j_3) \to \\
\mathit{enterBooth}.v \to \\
\bigsqcap_{(i_4, j_4) \in F1} \left(\begin{array}{l}
\mathit{place}.v.(i_4, j_4) \to \\
\bigsqcap_{(i_5, j_5) \in F2} \left(\begin{array}{l}
\mathit{place}.v.(i_5, j_5) \to \\
\mathit{Voter}_1(v, \mathit{Rest}, \{s_1, s_2, s_3\}, n-1)
\end{array} \right)
\end{array} \right)
\end{array} \right)
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{Voter}_1(v, \mathit{aset}, \mathit{serials}, 0) \mathrel{\widehat{=}} \\
&\quad \bigsqcap_{s \in \mathit{serials}} \left(\begin{array}{l}
\mathit{receipt}.v.s?(i, j) \to \mathit{leaveBooth}.v \to \\
\mathit{closeElection} \to \mathit{STOP}
\end{array} \right)
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{Voter}_1(v, \mathit{aset}, \mathit{serials}, r) \mathrel{\widehat{=}} \\
&\quad \mathit{place}.v?(k, l) \to \\
&\qquad \mathit{Voter}_1(v, \mathit{aset} \setminus \mathsf{Row}(k), \mathit{serials}, r-1)
\end{aligned}
$$

Thus the process representing all voters is described by the interleaving of the voters as follows:

$$\mathit{Voters} \mathrel{\widehat{=}} \|_{v \in \mathcal{V}} \, \mathit{Voter}(v)$$

Election Authority Process. The authority (electoral official) in the polling station is responsible for authenticating voters on the channel *auth* and assigning the pre-printed multi-ballots (three unique serial numbers from the set $\mathcal{S}$) to the voters with the *alloc* events). Moreover, the serial number allocations are performed non-deterministically and it is ensured that the serial number allocated is never assigned to another mini-ballot. Because the last line of a mini-ballot is allocated for serial numbers, each is placed on the nth column, where n is the number of candidates on the board (note that the row numbers range between 0 and n, thus ensuring that the serial numbers are always allocated at the bottom of the ballot forms). Additionally, the allocation occurs three times for each multi-ballot form, and once three are allocated, the remaining serial numbers are carried forward to the next voter. When the election is closed through the *closeElection* event, no more ballots are allocated for any voter.

$$
\begin{aligned}
&\mathit{Authority} \mathrel{\widehat{=}} \mathit{openElection} \rightarrow \mathit{Authority}_1(\mathcal{S})\\
&\mathit{Authority}_1(\mathit{serials}) \mathrel{\widehat{=}}\\
&\qquad \mathit{auth}?v \rightarrow \bigsqcap_{s \in \mathit{serials}} \left(\begin{array}{l} \mathit{alloc}.v.s.(n,0) \rightarrow \\ \mathit{Authority}_2(v,(n,0),\mathit{serials} \setminus \{s\}) \end{array} \right)\\
&\mathit{Authority}_2(v,\mathit{coord},\emptyset) \qquad\quad \mathrel{\widehat{=}} \mathit{closeElection} \rightarrow \mathit{STOP}\\
&\mathit{Authority}_2(v,(n,2),\mathit{serials}) \mathrel{\widehat{=}} \mathit{Authority}_1(\mathit{setSerials})\\
&\mathit{Authority}_2(v,(n,i),\mathit{serials}) \mathrel{\widehat{=}}\\
&\qquad \bigsqcap_{s \in \mathit{serials}} \Big(\mathit{alloc}.v.s.(n,i+1) \rightarrow \mathit{Authority}_2(v,(n,i+1),\mathit{serials} \setminus \{s\}) \Big)
\end{aligned}
$$

The Bulletin Board Process. The process *B_Board* operates as a bulletin board where the cast mini-ballots are published. In detail, the votes are collected by this process as the voters cast their mini-ballots and a record is kept of the serial numbers and the bubbles that are filled in on that particular mini-ballot with the set *bag*. Once a serial number is allocated to a coordinate (i,j) for any voter, the process traces the jth column, whenever a *place* event happens on that column, because the event *place*.$v.(i,j)$ models a vote for the ith candidate on the candidate list. Subsequently, the value i is then stored in the set *bag*. Additionally, the process is also ready to give any of the mini-ballots as the voter's receipt, with the serial number and bubbles filled in, if requested by them. After the election, the cast mini-ballots are published with the *pub* event. The whole process *B_Board* is modelled as the parallel composition of individual mini-ballot processes as follows.

$$
\begin{aligned}
&Board(s) \mathrel{\widehat{=}} alloc?v.s?(i,j) \rightarrow Board_1(\emptyset, s, (i,j))\\
&Board_1(bag, s, (i,j)) \mathrel{\widehat{=}}\\
&\qquad \mathop{\Box}_{(m,n)\in \mathsf{Col}(j)} \Big(place.v.(m,n) \rightarrow Board_1(bag \cup \{m\}, s, (i,j)) \Big)\\
&\qquad \Box\ receipt?v.s.bag \rightarrow Board_2(s, bag)\\
&\qquad \Box\ Board_2(s, bag)\\
&Board_2(s, bag) \mathrel{\widehat{=}} closeElection \rightarrow pub.s.bag \rightarrow bagempty \rightarrow STOP
\end{aligned}
$$

$$B\_Board \mathrel{\widehat{=}} openElection \rightarrow \|_{s\in\mathcal{S}} Board(s)$$

Counter Process. The last process is *Counters*, which works as an electoral official counting the votes published on the bulletin board, keeping a record of *place* events for each candidate by following each row of ballot forms. That is, when there exists a mark on the ith row of a mini-ballot, it is counted as a vote for the candidate c_{i+1}. Moreover, when no more *place* events are happening, the number of total votes for each candidate is published on the channel *total*.

$$
\begin{aligned}
&Counter(c_x, r) \mathrel{\widehat{=}}\\
&\qquad \mathop{\Box}_{\substack{v\in\mathcal{V}\\(i,j)\in\mathsf{Row}(x-1)}} \Big(place.v.(i,j) \rightarrow Counter(c_x, r+1) \Big)\\
&\qquad \Box\ bagempty \rightarrow total.c_x.r \rightarrow STOP
\end{aligned}
$$

$$Counters \mathrel{\widehat{=}} \|_{c\in\mathcal{C}} Counter(c, 0)$$

System Process. The ThreeBallot voting system model is the parallel composition of the processes defined previously. Hence, the composition is defined as follows:

$$System_{3B} \mathrel{\widehat{=}} Voters \parallel Authority \parallel Booth \parallel B\_Board \parallel Counters$$

3 Automated Anonymity Verification

Our analysis of ThreeBallot uses the formal anonymity definition given in [12]. The definition of anonymity for the voting systems, also called weak anonymity, is based on trace equivalence ($\equiv_T$) and expressed as follows:

Definition 1. *The process P is weakly anonymous on a set of channels C of type T if:*

$$P[\![{}^{c.x,\, d.x}/_{d.x,\, c.x} \mid x \in T]\!] \equiv_T P$$

for any $c, d \in C$

That is, when the two channels $c.x$ and $d.x$ are swapped over for all values of x, if the resulting process is indistinguishable in terms of traces semantics from the original process, P, from an observer's point of view, then the process provides anonymity.

The abstraction methods that are frequently used in the analysis are: the *hiding* abstraction method used as $P \setminus A$ to make occurrences of events in A internal, and hence invisible to an observer and the *renaming* method shown as $P[\![R]\!]$ for a relation R, so that the occurrences of an event a are replaced by events b such that aRb. A substitution-like notation is often used for describing relations. Regarding this, $P[\![{}^a/_b]\!]$ means that the event or channel b is replaced by a in P, e.g., $(b \rightarrow STOP)[\![{}^a/_b]\!] \mathrel{\widehat=} a \rightarrow STOP$, and $(b?x \rightarrow STOP)[\![{}^a/_b]\!] \mathrel{\widehat=} a?x \rightarrow STOP$. More generally, multiple substitutions, including $P[\![{}^{a,\, b}/_{b,\, a}]\!]$ (a maps to b and b maps to a), many-to-one renaming, $P[\![{}^{a,\, a}/_{b,\, c}]\!]$ (b and c both map to a) and one-to-many renaming $P[\![{}^{b,\, c}/_{a,\, a}]\!]$ (a maps to both b and c) are allowed.

It is over channel *choose* that the voter determines a choice of candidate; consequently, the channels that need to be swapped over are: $choose.v_1.c_x$ and $choose.v_2.c_x$ for $c_x \in candidates$. Therefore, the anonymity specification, denoted as Spec, for ThreeBallot CSP model ($System_{3B}$) can be described using the renaming operator as:

$$Spec \mathrel{\widehat=} System_{3B}[\![{}^{choose.v_1.c_x,\, choose.v_2.c_x}/_{choose.v_2.c_x,\, choose.v_1.c_x}]\!]$$

This is the left hand side of the trace equivalence in the definition above. As the anonymity property of the system is checked from an observer's point of view, the observer's inability to see sensitive information is extremely important. He is able to see all the public channels, but not the private channels: *alloc* and *place*. Therefore, these private channels need to be hidden, which is done by using the hiding operator.

$$Abs \mathrel{\widehat{=}} System_{3B} \setminus \{| \, alloc, place \, |\}$$

The resulting process, denoted Abs, is the abstracted CSP model of ThreeBallot. According to the definition above, if the two systems (abstracted model and specification) are trace equivalent then the system provides anonymity. Hence, we will check whether the equation, $Spec \equiv_{\mathrm{T}} Abs$, holds.

We assume that the adversary in our model is able to see all *receipt* events; i.e., he can see all the receipts taken in an election. (This is a strong assumption; however, if the system is secure under this assumption, it will also be secure with an adversary who sees only some receipts.)

3.1 Results for the ThreeBallot model with no SBA

Unsurprisingly, the previous trace equivalence does not hold for our ThreeBallot voting system model. This is because there are situations in which a reconstruction attack is possible: that is, a coercer who has seen receipts for v_1 and v_2 can infer that they voted respectively for c_1 and c_2 because there is no way of constructing a complete set of valid multi-ballots in which v_1 and v_2 vote for c_2 and c_1 respectively. Whether the election run provides anonymity entirely depends on how the voters fill their multi-ballots, and also on which mini-ballots they choose as receipts.

The following counter-examples from different voting scenarios give useful intuition about in what situations anonymity is not satisfied.

Examples of Privacy Violations of ThreeBallot.

Example 1. The first counter-example is taken from a protocol run with two voters, v_1 and v_2, and two candidates, c_1 and c_2. The FDR2 model checker returns several counter-examples that violate anonymity (we discuss the size and computational complexity of the models and verification times in Sect. 5). We examine one of these traces here, illustrating the receipts taken by the voters and the mini-ballots displayed on the bulletin board. The following illustrated examples are the election runs from the observer's point of view.

The counter-example trace shows that in a voting scenario as in Fig. 7, where v_1 chooses to vote for c_1, and v_2 votes for c_2, if the voters take s_2 and s_3 respectively as their receipts, the observer is able to reconstruct the multi-ballots from the public mini-ballots on the bulletin board. There is no possible reconstruction where the votes were cast the other way round. Therefore, the observer is able to say who voted for whom in this ThreeBallot election run.

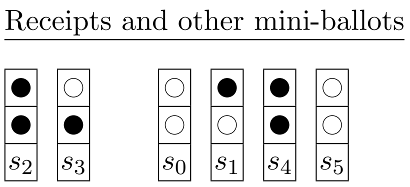

Fig. 7. Voting scenario 1.

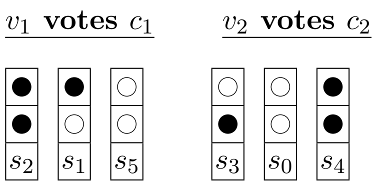

Fig. 8. Reconstruction attack 1.

With the public information shown on the bulletin board and the receipts that the voters share with the coercer, the only way of reconstructing these votes is illustrated in Fig. 8. The mini-ballots s_0 and s_5 can be swapped. However, it does not affect the way the voters have voted.

Example 2. In an election with three voters and two candidates, as depicted in Fig. 9, when voter v_1 votes for c_1, voter v_2 votes for c_2, and voter v_3 votes for c_1, with the receipts s_1, s_2 and s_0 respectively, voter v_1 can be seen not to have voted for c_2. Figure 10 shows the only possible reconstruction.

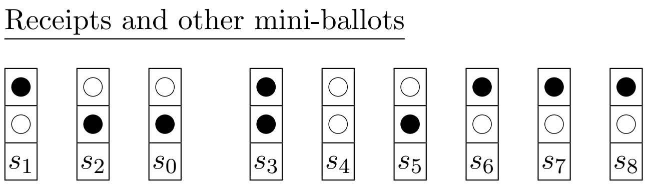

Fig. 9. Example 2. voting scenario

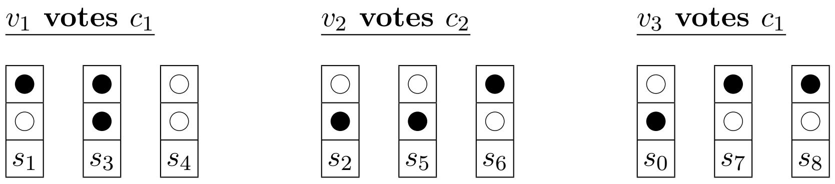

Fig. 10. Example 2. reconstruction attack

3.2 Short Ballot Assumption

We now analyse the ThreeBallot voting system under two of the three possible interpretations of the SBA that were given earlier: Assumptions 2 and 3. (Recall that Assumption 1 seemed implausible unless there were only very few candidates.)

Analysis under the SBA-mini. Suppose we adopt Assumption 2, under all possible mini-ballots are assumed to appear on the bulletin board at least once at the end of the election. We give here a simple counter-example to show that ThreeBallot does not provide anonymity. In the example in Fig. 11, receipt s_0 has two possible completions: it could be combined with s_2 and s_4 or s_8 (as depicted in Fig. 12), or with s_5 and s_7. But in either case, it represents a vote for the third candidate.

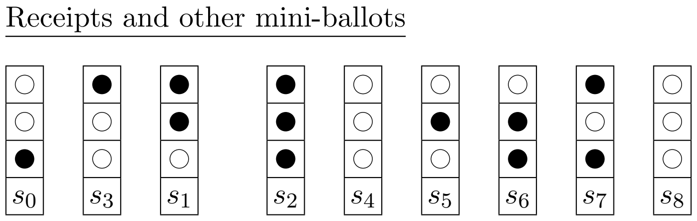

Fig. 11. An example voting scenario: all possible mini-ballots appear on the bulletin board

Analysis under SBA-mini-t. Suppose now that we adopt Assumption 3, which ensures that every possible mini-ballot will appear on the bulletin board at least t times for some suitable value of t. We show here that this is insufficient regardless of the value of t.

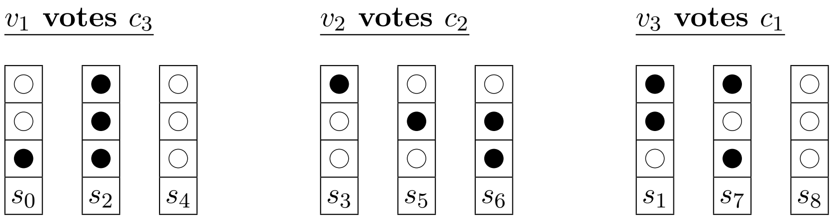

Fig. 12. Reconstruction attack

We start by observing that a fully filled mini-ballot can be combined only with an empty mini-ballot and a singleton. Additionally, any possible mini-ballot m that is not empty, fully filled or a singleton can be turned into a completed multi-ballot that does not contain a fully filled mini-ballot or a singleton. This can be done by combining it with another mini-ballot that is the complement of m but with one extra bubble, and an empty mini-ballot.

We can reach a bulletin board that displays at least t copies of every possible mini-ballot in the following way. For each possible mini-ballot that is not empty, fully filled or a singleton, we turn it into a multi-ballot as described above, and add it to the board. This gives us at least t copies of everything except singletons and fully filled mini-ballots.

Now each possible singleton should be combined with a fully filled mini-ballot and an empty mini-ballot. We add t copies of each such multi-ballot to the board. This means that every possible mini-ballot now appears at least t times.

However, any voter who takes a singleton as a receipt will not have any hope of anonymity. The number of fully filled mini-ballots is the same as the number of singletons; and since each fully filled ballot must be combined with a singleton and a blank, it follows that the voter's receipt must have been part of such a multi-ballot. But in that case the mini-ballot reveals the candidate that the voter selected. Hence no value of t is sufficient to guarantee anonymity in ThreeBallot.

SBA-pro: A better formulation. We have seen that the interpretations of the SBA given so far are either not enough or unrealistic. We now give a much more plausible short ballot assumption that is demonstrably strong enough for ThreeBallot.

Assumption 4 (SBA-pro) *Let M be the set of all mini-ballots that were cast during the election; $R \subset M$ is the set of all of the receipts that are known to the adversary. We introduce a partial function vote such that $vote(m_1, m_2, m_3) = c$ whenever the three mini-ballots m_1, m_2 and m_3 together form a valid multi-ballot that represents a vote for c. Additionally, for any two mini-ballots m_1 and m_2, we say that $m_1 \sim m_2$ if and only if they contain the same sequence of vote marks (that is, $m_1 = m_2$ except possibly for the serial numbers).*

For every $r \in R$ and every candidate c, there was a vote cast consisting of three (unordered) mini-ballots m_1, m_2, m_3 such that

1. $r \sim m_1$;
2. $vote(m_1, m_2, m_3) = c$;
3. $m_2, m_3 \in M \setminus R$.

Informally, this interpretation says that for every receipt known to the adversary, there was an equivalent one used in a multi-ballot for each of the candidates in the election.

Theorem 1. *Assumption 4 is strong enough to prevent reconstruction attacks in ThreeBallot.*

Proof. The key to the proof is the observation that if $m \sim m'$ then we must have $vote(m, m_2, m_3) = vote(m', m_2, m_3)$. This is clear from the fact that m and m' can differ only in their serial numbers, and the serial numbers are not relevant for determining which candidate received the vote cast by a multi-ballot.

Suppose that $r \in R$, and the adversary wishes to determine which candidate received the vote cast that included r. We can see that any candidate is possible. Suppose that r did in fact occur in a multi-ballot along with m_1 and m_2, as a vote for c. For any other candidate c', there was a multi-ballot cast containing m_3, m_4, m_5 such that $vote(m_3, m_4, m_5) = c'$ and $r \sim m_3$, and with m_4 and m_5 not known to the adversary.

But this means that the adversary cannot distinguish the following two possibilities:

1. a ballot of (r, m_1, m_2) for c, and a ballot of (m_3, m_4, m_5) for c';
2. a ballot of (m_3, m_1, m_2) for c, and a ballot of (r, m_4, m_5) for c'.

In each case, the set of mini-ballots used by this partial reconstruction is the same, so it cannot affect further reconstruction of the remaining mini-ballots. In one case, r was used to vote for c, and in another case, for c'; and since c' was arbitrarily chosen, we conclude that r could equally have been used to vote for any candidate.

To see the improved plausibility of this interpretation, suppose the adversary has knowledge of r receipts in an election run with n candidates. SBA-pro requires at least $n \cdot r$ multi-ballots of the right type to have been cast to protect anonymity. By contrast, SBA-multi requires at least $n \cdot 3^n$ other appropriate multi-ballots. As long as r is small, SBA-pro is much less demanding compared with SBA-multi. For instance, in an election with 10 candidates, SBA-multi needs at least 590,490 multi-ballots. Unless the adversary has seen somewhere in the order of 59,000 receipts, SBA-pro is much more likely to be satisfied.

This efficiency argument is compelling, but not absolute: to formalise it would require a full voter model; that is, it would need a probability distribution over multi-ballots cast in the election. Producing such a model is probably unrealistic, since it would be affected by the prevailing political landscape at the time of the election; it is in any case outside the scope of this paper.

3.3 Verified Privacy Cases

Apart from the short ballot assumption, several slight modifications of ThreeBallot have been proposed to help the system provide absolute anonymity. Using FDR we were able to verify these modified systems against reconstruction attacks with a limited number of voters, candidates and ballot forms. We have automatically verified a ThreeBallot model that allows voters to exchange their receipts; and we analyse the system with an additional constraint that voters must fill in at least one bubble in every column.

Floating/Exchanging Receipts. Rivest [5] suggests a possible improvement to the original ThreeBallot scheme with the idea of exchanging receipts in the polling station. Each voter puts her receipt in a box, and takes someone else's receipt. Indeed, this idea can be used in any paper-based election system. If we let voters take a random receipt from the box in the polling station, then this eliminates reconstruction attacks as well as pattern-matching (Italian) attacks because the adversary does not have any knowledge of any part of the voter's ballot. Although the adversary may be able to reconstruct valid multi-ballots, he cannot link them to voters. We have verified using FDR that the modified scheme, where the voters are allowed to exchange their receipts, provide guaranteed anonymity.

No Single Mini-ballot Left Blank. We here add a condition that voters must fill out at least one bubble on each mini-ballot. For the two candidate case, there are only two ways of filling a mini-ballot, and thus only two different receipt that can be taken by voters. We have modified our model to provide automatic verification that this condition is sufficient to guarantee anonymity with two candidates. However, in an election where there are more candidates than two, although intuitively the system provides better probabilistic anonymity than the original, it cannot guarantee voter anonymity.

4 VAV

VAV is a modification of ThreeBallot proposed by Rivest and Smith [6]. The voting ceremony is the same as in ThreeBallot. However, the ballot papers in VAV are a little different. As seen in Fig. 13 each mini-ballot has either "V" or "A", which stand for "Vote" and "Anti-vote" respectively, printed across the top. Two mini-ballots that have the same pattern but different types (one V and the other A) cancel each other out when counted. To vote for a candidate the voter fills the bubbles for the same random candidate in one of the V and the A mini-ballots, and then expresses her preference (her real vote) with the third mini-ballot marked as V. For instance, in Fig. 13, the voter marks the first two mini-ballots for Alice, and the third for Bob. In ThreeBallot the voter needs to make exactly one or two marks for each candidate on a multi-ballot (row-constraints); in VAV, however, there is no such need. As in ThreeBallot, the voter chooses one of the mini-ballots as her receipt and casts each mini-ballot separately into the ballot box. After the election, all mini-ballots are published on a bulletin board so the voters can verify that at least one of their three mini-ballots has been cast and included in the final tally. In the counting phase, the election officials remove the matching V and A mini-ballots and count the mini-ballots that are left, all of which should be V mini-ballots. It is claimed in [6] that the VAV protocol provides anonymity (vote privacy) under a version of the SBA. With VAV in an n-candidate race, the number of possible mini-ballot patterns is $2n$, which is small and easily satisfies the SBA.

V		A		V	
Alice	●	Alice	●	Alice	○
Bob	○	Bob	○	Bob	●
Chris	○	Chris	○	Chris	○
David	○	David	○	David	○
	65375		10835		40193

Fig. 13. A VAV multi-ballot, filled in as a vote for Bob

4.1 Modelling VAV in CSP

As the VAV voting system is similar in structure to ThreeBallot, modelling of VAV in CSP does not require much by way of modification of the ThreeBallot model. However, the following changes are needed in order to achieve correct behaviour. First, as the mini-ballots in VAV contain V and A indicators on them, we reflect this information in the model by defining V and A datatypes in order to distinguish two mini-ballots with the same pattern, but with different types (Vote, Anti-Vote). Secondly, in order to pass such information between channels, the channels used in ThreeBallot (*receipt*, *alloc* and *pub*, and so on) are modified.

In terms of processes, the *Voter*() process is modified as the voter behaviours in ThreeBallot and VAV are slightly different when filling the ballot forms. In VAV model, the voter first places a mark for her chosen candidate, then mark a V and A mini-ballot in such a way that they cancel each other out. In addition, the voter does not have to fill a bubble for each candidate. The authority and bulletin board processes are the same as in ThreeBallot. The counter process differs in how it counts votes: in ThreeBallot, the counter process counts the marks performed by *place* events, but in VAV, the counter process counts what has been published on the BB by keeping track of *pub* events in an election run.

4.2 Automated Verification of VAV

In this section, we present automated verification of VAV against the same anonymity specification given in Sect. 3. The following are the key observations made during the automated analysis with FDR.

Observation 1. When the mini-ballots that appear on the BB and are not taken as receipts are all the same (having the same pattern — see the mini-ballots on the BB in the following example), the receipts, all of which are known to the intruder, will show how the voters have voted. For instance,

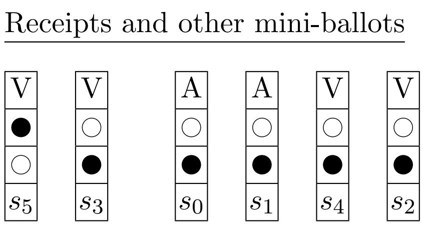

Fig. 14. Voting scenario 1.

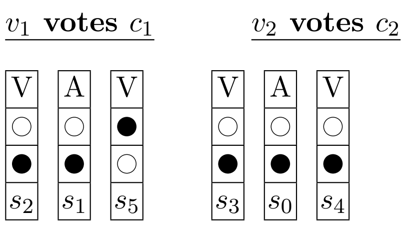

Fig. 15. Reconstruction attack 1.

Observation 2. When there is a vote for a candidate, say c, there should be enough mini-ballots marked for c that were not taken as receipt. Otherwise, no one can have voted for c except the voter holding a receipt marked for c.

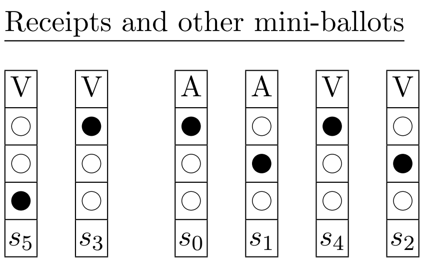
Receipts and other mini-ballots
V V A A V V
s_5 s_3 s_0 s_1 s_4 s_2

Fig. 16. Voting scenario 2.

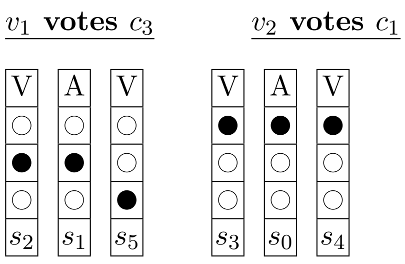

Fig. 17. Reconstruction attack 2.

In Fig. 16 and 17, for instance, looking at the receipts and the final outcome (1 vote for c_3 and 1 for c_1), voter v_1, who holds the receipt s_5, cannot have voted for c_1 as there is no Anti-vote mini-ballot marked for c_3 that can cancel out her receipt. Similarly, v_2 cannot have voted for c_3, as there is no spare mini-ballot marked for c_3 that was not taken as a receipt. Hence, v_2 has voted for c_1.

Observation 3. In addition the previous observations, we note that the notion of *having the same pattern* must now include having the same type (V or A). A voter who holds a receipt marked as 'A' will need there to be a mini-ballot marked as 'V' in order to cancel it out. The following example illustrates this point.

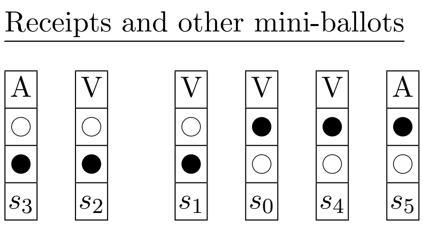
Receipts and other mini-ballots
A V V V V A
s_3 s_2 s_1 s_0 s_4 s_5

Fig. 18. Voting scenario 3.

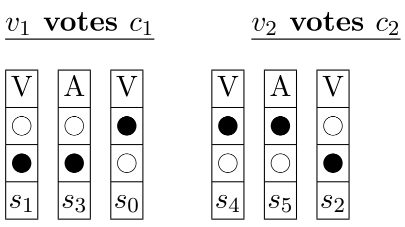

Fig. 19. Reconstruction attack 3.

Figure 18 and 19 show that voter v_1 has voted for c_1 and chose the mini-ballot s_3 as her (Anti-vote) receipt; similarly, voter v_2 has voted for c_2 and chose the mini-ballot s_2 as her receipt. The counter example is produced by FDR once the mini-ballot with the serial number s_5 is published on the BB. An Anti-vote requires a mini-ballot (Vote) with the same pattern; thus, the Anti-vote s_3 requires the existence of the mini-ballot s_1, and similarly the published mini-ballot s_5 requires a mini-ballot like s_4 or s_0. As there is only one vote for each candidate, it is easy to check whether v_1 with the receipt s_3 can ever vote for c_2. As s_1 and s_3 are the two mini-ballots of the multi-ballot cast by v_1, which cancel each other out, there must be a spare mini-ballot (not taken as receipt) on the BB marked for the candidate c_2, so that v_1 could vote for c_2. However, as the only such mini-ballot is s_2, which is the receipt of voter v_2, voter v_1 cannot have voted for c_2. On the other hand, if voter v_1 has taken mini-ballot s_1, which has the same pattern as her receipt but marked as V, she could have voted for either of the candidates.

4.3 Short Ballot Assumption for VAV

We now analyse the VAV voting system under two of the interpretations of the SBA given earlier: Assumptions 2 (SBA-mini) and 3 (SBA-mini-t). Once we have shown that these are inadequate, we will give a better formulation that is both realistic and sufficient to guarantee anonymity.

Analysis under SBA-mini. There are $2n$ possible mini-ballots in VAV, where n is the number of candidates. We consider mini-ballots that have the same pattern but different types as having different patterns, unlike the authors in [6], who, when discussing the SBA for VAV, say that the number of possible mini-ballot patterns is n, and so presumably do not consider different types to imply different patterns. Since the mini-ballot types are visibly different (and must be so in order to tally the election), it seems natural to treat these types as different patterns.

The version of the SBA under consideration here requires each mini-ballot to be cast at least once. The example in Fig. 18 and 19 shows an attack under this version of the SBA, automatically produced by FDR. In the example, all $2n$ possible mini-ballots appear on the BB, but there is still a violation of privacy.

Analysis under SBA-mini-t. Here we show that reconstructions are still possible even when each mini-ballot is required to be cast at least t times, regardless of the value of t. Let us suppose that we have two candidates. We will ensure that all possible mini-ballots appear on the BB at least t times by having $2t$ voters, t of whom vote for c_1, and the others for c_2, as exemplified in Fig. 20. All mini-ballots appear on the BB at least t times; however, anonymity is not provided by the VAV protocol. Those that voted for c_2 could not have voted for c_1, and those that voted for c_1 could not have voted for c_2, as there are not enough spare A mini-ballots that were not taken as receipts. (Observe in particular that all receipts of type V that show a mark for c_2 must represent a vote for c_2 unless they have been cancelled with a similar mini-ballot of type A; but all such mini-ballots have been taken as receipts by others.)

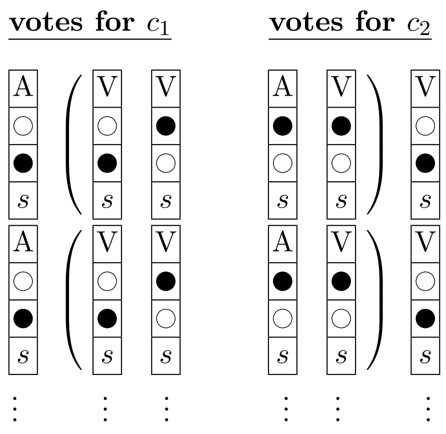

Fig. 20. SBA-mini-t Counter Example: All the mini-ballots shown are published on the BB, and the mini-ballots outside the parentheses represent the receipts taken by the voters and known to the intruder.

We now turn to a reading of the SBA that is highly plausible and also sufficient for anonymity.

Let us suppose that a voter v has cast a vote for candidate c. We want to ensure that an attacker cannot distinguish this situation from one in which this was a vote for a different candidate. The most natural approach to providing anonymity is to ensure that there is another voter v', this time voting for c', such that an attacker cannot distinguish this case from the one in which v voted for c' and v' voted for c. In other words, given the receipt that v and v' received, and the four mini-ballots that made up the rest of these multi-ballots, there is one reconstruction in which v voted for c and v' for c', and another reconstruction where v voted for c' and v' for c.

There are three possibilities for the type of receipt that v might have received: an antivote; a vote that was cancelled with an antivote; and a real vote. (If v chose the same candidate on all three mini-ballots then these last two are indistinguishable, but for what follows, the two mini-ballots of type V can be arbitrarily designated one as the real vote and one as the cancelled vote.)

If v took an antivote, then the multi-ballot is of the form $A(x)$, $V(x)$, $V(c)$ for some x, where $A(q)$ represents an antivote for q, and $V(q)$ represents a vote. The first-listed component is the receipt; the order of the other two is unimportant. This, then, contains a cancelled vote for x, and a real vote for c; it is the antivote that was taken as the receipt.

In order to mask this vote, we now need a vote of the form $A(y)$, $V(y)$, $V(c')$ for some y, or $V(y)$, $A(y)$, $V(c')$. In other words, it can be masked with any vote for c' where the real vote was not taken as a receipt. The attacker cannot distinguish between this case, and the case where $V(c)$ and $V(c')$ were swapped over.

In order to ensure that v could have voted for any other candidate, we will therefore need, for each candidate b, the existence of any vote for b that does not take the real vote as a receipt.

If v took $V(x)$ (the vote that was cancelled by the antivote) as a receipt, exactly the same masking vote will suffice.

The hardest case is where the voter takes the real vote as a receipt—that is, the vote is of the form $V(c)$, $A(x)$, $V(x)$. In order to mask this, and make it possible that v voted for c', the $V(c)$ will need to be cancelled out, so the masking vote will need to contain an $A(c)$ in it, and this $A(c)$ cannot have been taken as a receipt (or else the attacker will know it was not part of v's vote). The masking vote must also contain a $V(c)$ in it, since it has an A(c). That leaves us with two possibilities: the masking vote is either $V(c')$, $A(c)$, $V(c)$ or else it is $V(c)$, $A(c)$, $V(c')$. (These two are the same, except for choice of receipt.)

The former will not work, because the $V(c')$ will need to be plausibly part of v's vote, as far as the attacker is concerned, if v is plausibly to have voted for c'. So the masking vote must be of the form: $V(c)$, $A(c)$, $V(c')$.

This leaves us with two votes, one cast by v, and of the form $V(c)$, $A(x)$, $V(x)$; and one cast by v', of the form $V(c)$, $A(c)$, $V(c')$. This, for an attacker, is indistinguishable from the case where v casts $V(c)$, $A(c)$, $V(c')$, and v' casts $V(c)$, $A(x)$, $V(x)$.

In order to ensure that v could plausibly have voted for any candidate, we will therefore need, for each candidate z, a vote for z that took a receipt as $V(c)$.

This suggests a neat simplification. The reason that the masking worked here is that the receipts that v and v' took are the same. Regardless of the form of the vote, v will maintain anonymity as long as, for any candidate z, someone cast a vote for z but took the same receipt as v did. We are left, then, with the following reading of the SBA for VAV elections.

Assumption 5 (SBA-pro-VAV) *For every possible mini-ballot m, and for every candidate z, at least one voter casts a vote for z and takes a receipt with the same pattern as m.*

If there are n candidates, then there are $2n$ different possible receipt patterns, so the total number of receipt-candidate pairings that must turn up is $2n^2$. This seems eminently reasonable: even with, say, 10 candidates, which would usually indicate a very large election, there are only 200 receipt-candidate pairings that must appear in order to ensure that all voters maintain their anonymity.

5 Conclusion

In this paper, we have demonstrated that the ThreeBallot and VAV voting systems are vulnerable to privacy-related attacks, especially reconstruction attacks, even under some plausible interpretations of the short ballot assumption.

In our analysis, we have used abstracted CSP models of ThreeBallot and VAV, which are defined as the parallel composition of agents in the system. We model the adversary in the analysis as an outsider/observer, who can see all the public channels, including what each voter takes as a receipt. We have given a number of examples for different voting scenarios, demonstrating that ThreeBallot and VAV do not provide anonymity under various formulations of the short ballot assumption. We have in addition given reasonable and plausible interpretations of the short ballot assumption for ThreeBallot and VAV separately that do in fact prevent reconstruction attacks. Additionally, we have considered two different versions of ThreeBallot that we were able to analyse automatically using FDR; namely, exchanging receipts and no single mini-ballot left blank.

Because of the state space limitation that all model checking tools suffer from, we were able to analyse the models with a limited number of agents. In most cases, the restriction did not affect the analysis of the systems and assumptions; however, as the short ballot assumptions require a large number of mini-ballots, we were not able to demonstrate automatic verification in such cases; however, we have supplied hand proofs where appropriate. Moreover, although automated analysis has been shown to be a successful approach as proved by finding counter-examples for a failure of the protocol requirement, proving directly that the system meets the claimed properties would need an infinite-state model. In order to generalise the verification to models of arbitrary size, there are several techniques in the literature that can be employed, such as *structural* and *data-independent* induction [23, 24, 25]. However, data-independence techniques do not easily apply to the models that have been developed here as the established results require rather strict conditions, which have not been satisfied in our models. Similarly, the structural induction technique also appears to be a promising approach, but there are a few limitations with this technique too. In order to give an idea to the reader about the verification times, automated analysis was conducted on a machine with Intel® Core™ i5 CPU 2.40GHz, and 1GB RAM, see Table 1 ("–" means no result is produced in a reasonable time). In the table, the verification times of the CSP models of the ThreeBallot versions are illustrated, respectively; the original ThreeBallot, the ThreeBallot version where there is no mini-ballot left empty (*No mini-b empty*) when filling a ballot form, and the version where all possible mini-ballots appear on the bulletin board (*All mini-b appear*).

Table 1. FDR verification times for ThreeBallot versions

	Original		No mini-b empty		All mini-b appear	
	States	Time	States	Time	States	Time
2v 2c	239, 905	7.8s	56, 841	5.3s	240, 055	7.0s
2v 3c	4, 139, 347	1m41.8s	1, 435, 926	38.3s	4, 165, 428	1m40.1s
3v 2c	–	–	67, 409, 391	22m49.3s	–	–
3v 3c	–	–	–	–	–	–

Acknowledgements

Some of the work was conducted while authors were at the University of Surrey and carried out under the EPSRC-funded trustworthy voting systems (TVS) project EP/G025797/1. First author's work is sponsored by The Ministry of Education Republic of Turkey.

References

[1] Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM **24** (February 1981) 84–90

[2] Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: AUSCRYPT. (1992) 244–251

[3] Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. IACR Cryptology ePrint Archive **2002** (2002) 165

[4] Chaum, D., Ryan, P.Y.A., Schneider, S.A.: A practical voter-verifiable election scheme. In: ESORICS. (2005) 118–139

[5] Rivest, R.L.: The Threeballot voting system (2006)

[6] Rivest, R.L., Smith, W.D.: Three voting protocols: ThreeBallot, VAV, and Twin. In: Proceedings of USENIX/ACCURATE Electronic Voting Technology (EVT), Press (2007)

[7] Hoare, C.A.R.: Communicating sequential processes. Communications of the ACM **21** (August 1978) 666–677

[8] Gardiner, P., Goldsmith, M., Hulance, J., Jackson, D., Roscoe, B., Scattergood, B., Armstrong, B.: FDR2 user manual

[9] Backes, M., Hritcu, C., Maffei, M.: Automated verification of remote electronic voting protocols in the applied pi-calculus. In: CSF. (2008) 195–209

[10] Smyth, B.: Formal verification of cryptographic protocols with automated reasoning. PhD thesis, School of Computer Science, University of Birmingham (2011)

[11] Ryan, P.Y.A., Schneider, S.A.: Prêt à Voter with re-encryption mixes. In: ESORICS. (2006) 313–326

[12] Moran, M., Heather, J., Schneider, S.: Verifying anonymity in voting systems using CSP. Formal Aspects of Computing (2012) 1–36

[13] Cichoń, J., Kutylowski, M., Weglorz, B.: Short ballot assumption and Threeballot voting protocol. In: Proceedings of the 34th conference on Current trends in theory and practice of computer science. SOFSEM'08, Berlin, Heidelberg, Springer-Verlag (2008) 585–598

[14] de Marneffe, O., Pereira, O., Quisquater, J.J.: Simulation-based analysis of E2E voting systems. In: Proceedings of the 1st international conference on E-voting and identity. VOTE-ID'07, Berlin, Heidelberg, Springer-Verlag (2007) 137–149

[15] Strauss, C.: The trouble with triples: A critical review of the triple ballot (3ballot) scheme part1 (2006) Available at http://www.cs.princeton.edu/appel/voting/Strauss-TroubleWithTriples.pdf.

[16] Strauss, C.: A critical review of the triple ballot voting system, part2: Cracking the triple ballot encryption (2006) Available at http://www.cs.princeton.edu/appel/voting/Strauss-ThreeBallotCritique2v1.5.pdf.

[17] Clark, J., Essex, A., Adams, C.: On the security of ballot receipts in E2E voting systems. In: IAVoSS Workshop On Trustworthy Elections (WOTE). (july 2007)

[18] Appel, A.W.: How to defeat Rivest's ThreeBallot voting system. Unpublished (2007)

[19] Tjøstheim, T., Peacock, T., Ryan, P.Y.A.: A case study in system-based analysis: The ThreeBallot voting system and Prêt à Voter. In: VoComp. (2007)

[20] Henry, K., Stinson, D.R., Sui, J.: The effectiveness of receipt-based attacks on ThreeBallot. Trans. Info. For. Sec. **4**(4) (December 2009) 699–707

[21] Küsters, R., Truderung, T., Vogt, A.: Verifiability, privacy, and coercion-resistance: New insights from a case study. In: Security and Privacy (SP), 2011 IEEE Symposium on. (May 2011) 538 –553

[22] Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: Proc. 42nd IEEE Symp. Foundations of Computer Science. (2001) 136–145

[23] Roscoe, A.W.: The Theory and Practice of Concurrency. Prentice Hall PTR, Upper Saddle River, NJ, USA (1997)

[24] Lazic, R.S.: A Semantic Study of Data Independence with Applications to Model Checking. D. phil. thesis, Oxford University Computing Laboratory (1999)

[25] Roscoe, A.W.: Understanding Concurrent Systems. 1st edn. Springer-Verlag New York, Inc., New York, NY, USA (2010)