Automated Anonymity Verification of the ThreeBallot Voting System

Murat Moran **, James Heather, Steve Schneider

University of Surrey, Guildford, 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 voting system is important because it aims to provide security (voter anonymity and voter verifiability) without requiring cryptography. In this paper, we construct a CSP model of ThreeBallot, and use it to produce the first automated formal analysis of its anonymity property.
Along the way, we discover that one of the crucial assumptions under which ThreeBallot (and many other voting systems) operates—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. Therefore, we give a version of the Short Ballot Assumption for ThreeBallot that is realistic but still provides a guarantee of anonymity.

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

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 voting system [5] is particularly interesting because it uses no cryptography, but nevertheless still aims to provide anonymity, integrity of the election, verifiability and incoercibility.

** Corresponding author's work is sponsored by The Ministry of Education Republic of Turkey, m.moran@surrey.ac.uk, tel: +44(0)1483 682263, fax: +44(0)1483 686051.
The final 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`

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 it is essential for preventing coercion and vote buying. This paper considers the anonymity property as it relates to the ThreeBallot voting system.

ThreeBallot relies 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 a formal model of ThreeBallot in Communicating Sequential Processes (CSP) [6], and use the Failures-Divergences Refinement (FDR2) model checker [7] 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 [8] with hand-proofs, FOO [2] in [9] with a compiler, and Prêt à Voter [10] in [11]); but the ThreeBallot voting system has 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 Section 2, we model ThreeBallot as a parallel composition of agents: voters, an authority, and a bulletin board. Then, using an anonymity definition given in [11], in Section 3.1 we analyse our model against an adversary who can observe all public channels. Initially, our model drops the SBA entirely, and we discover that FDR leads us to several attacks on vote anonymity. Section 3.2 then discusses the Short Ballot Assumption 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 Section 3.3 we analyse the other versions of ThreeBallot, and demonstrate that with the modifications, ThreeBallot provides guaranteed anonymity. Finally, the Section 4 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 [12].

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 Table 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. The

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

Alice	●	Alice	●	Alice	○
Bob	○	Bob	●	Bob	○
	56248		04578		31489

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 [12] to be secure under the short ballot assumption (SBA). Rivest and Smith in [12] 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 [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 Section 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-n) *Every possible mini-ballot will be cast at least n times (for some suitably chosen n).*

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 n.

1.2 Related Work

The ThreeBallot voting system has been subjected to analysis of one sort or another many times since its publication [15, 16, 17, 18, 14, 19, 13, 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, the definition of weak anonymity used in [13] is much different from ours given in [11]. 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.

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, we model the ThreeBallot voting system using CSP. We assume that the reader is familiar with CSP notation; for details see Roscoe's book [23].

2.1 Data-types, Functions and Sets

We treat the multi-ballot of the ThreeBallot voting system as a board with co-ordinates. Here, a co-ordinate (i, j) defines a bubble on a mini-ballot, which is to be filled in. Thus, we have exactly three columns representing three mini-ballots, and a number of rows, which is one more than the number of candidates (the last row is allocated just for serial numbers). The size of the board is determined by these parameters: the number of voters, *VTRS*, and the number of candidates, *CNDS*. These parameters define the sets of voters, candidates and serial numbers. The data-types for voters, candidates and serial numbers are defined as $v.i$, $c.j$ and $s.k$ respectively.

We need several functions, which return a specific part of the board. For instance, $Row(i)$ returns the ith row of a multi-ballot form and $Col(j)$ is the set of bubbles on the jth column of a multi-ballot. Likewise, some functions call back the neighbouring bubbles of a given coordinate. For example, the function $adjR(i, j)$ returns the coordinates adjacent to (i, j) in the same row, similarly $adjC(i, j)$ returns the coordinates adjacent to (i, j) in the same column, and $nhdAll(i, j)$ returns all the neighbours of (i, j) in the current multi-ballot coordinates.

2.2 Processes and Channels

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. Fig. 1 illustrates the network for the ThreeBallot CSP model.

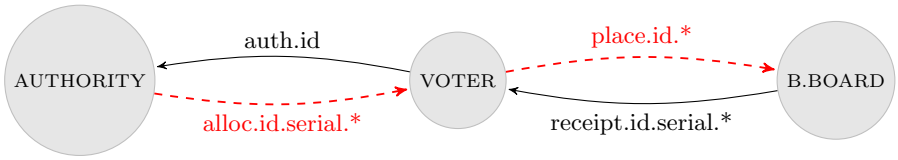


Fig. 1. ThreeBallot CSP Model Communication Channels ((⇢)private channel)

Voter Process. The voter chooses the candidate that she wants to vote for before the election is open. She then authorises herself with the election authority, and collects her multi-ballot with the *alloc* events. In the booth, the voter fills out two bubbles for the chosen candidate with the *place* events and one for the other candidates. 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 does *place* events in an efficient way; first a bubble from the first or second column is chosen for the candidate the voter wants to vote for then the second bubble is chosen from the other columns in a right to left fashion. Afterwards the process does one *place* event from top to bottom manner for the other candidates. The set $nhdAll(i,j) \setminus (Row(i) \cup Row(CNDS))$ is the set of bubbles left that can be filled in, and $CNDS$ is the number of candidates, which also identifies the number of rows.

$$
\begin{aligned}
VOTER(id) \mathrel{\widehat{=}} & \sqcap_{c.x \in candidates} choose!id.c.x \rightarrow openElection \rightarrow auth!id \rightarrow \\
& alloc.id?s1?(i1,j1) \rightarrow alloc.id?s2?(i2,j2) \rightarrow alloc.id?s3?(i3,j3) \rightarrow \\
& enterBooth!id \rightarrow \sqcap_{(i,j) \in Row(x-1) \setminus Col(2)} place!id.(i,j) \rightarrow \\
& \sqcap_{(i1,j1) \in adjR(i,j)} place!id.(i1,j1) \rightarrow \\
& VOTER'(id, nhdAll(i,j) \setminus (Row(i) \cup Row(CNDS)), \{s1,s2,s3\}, CNDS-1)
\end{aligned}
$$

$$
\begin{aligned}
VOTER'(id, aSet, setsers, 0) \mathrel{\widehat{=}} & \sqcap_{rcp \in setsers} receipt.id.rcp?(i,j) \rightarrow leaveBooth!id \rightarrow \\
& closeElection \rightarrow STOP
\end{aligned}
$$

$$
\begin{aligned}
VOTER'(id, aSet, setsers, cntr) \mathrel{\widehat{=}} & place.id?(k,l) \rightarrow \\
& VOTER'(id, aSet \setminus Row(k), setsers, cntr-1)
\end{aligned}
$$

Thus the process representing all voters is described by the parallel composition of the voters as:

$$VOTERS \mathrel{\widehat{=}} \|_{id} VOTER(id)$$

Election Authority Process. The election official in the polling station is responsible for authenticating voters with the events *auth* and assigning the pre-printed multi-ballots (three unique serial numbers for each voter) to the voters with an *alloc* event. The authority process is defined as follows:

$$AUTHORITY \mathrel{\widehat{=}} openElection \rightarrow AUTHORITY'(serials)$$

$$AUTHORITY'(setSrls) \mathrel{\widehat{=}} auth?id \rightarrow \bigsqcap_{srl \in setSrls} alloc.id.srl.(CNDS, 0) \rightarrow AUTHORITY''(id, (CNDS, 0), setSrls \setminus \{srl\})$$

$$AUTHORITY''(id, coord, \emptyset) \mathrel{\widehat{=}} closeElection \rightarrow STOP$$
$$AUTHORITY''(id, (CNDS, 2), setSerials) \mathrel{\widehat{=}} AUTHORITY'(setSerials)$$
$$AUTHORITY''(id, (CNDS, i), setSerials) \mathrel{\widehat{=}} \bigsqcap_{srl \in setSerials} alloc.id.srl.(CNDS, i+1) \rightarrow AUTHORITY''(id, (CNDS, i+1), setSerials \setminus \{srl\})$$

The authority opens the election, authorizes the voters, and assigns serial numbers to each mini-ballot with the *alloc* events. After the election, the authority performs *closeElection*, after which no more ballots can be allocated.

The Bulletin Board Process. The process *B_BOARD* operates as a bulletin board where the cast mini-ballots are published. The votes are collected while the voters cast their mini-ballots. Thus, the process keeps a record of the serial numbers and the bubbles that are filled in the set *Bag*. The mini-ballots are published with the *pub* event after the election is closed.

$$BOARD(srl) \mathrel{\widehat{=}} alloc?id!srl?(i, j) \rightarrow BOARD'(\emptyset, srl, (i, j))$$

$$BOARD'(Bag, srl, (i, j)) \mathrel{\widehat{=}} place.id?(m, n) : Col(j) \rightarrow BOARD'(Bag \cup \{m\}, srl, (i, j)) \\ \Box\ receipt?id!srl.Bag \rightarrow BOARD''(srl, Bag) \\ \Box\ BOARD''(srl, Bag)$$

$$BOARD''(srl, Bag) \mathrel{\widehat{=}} closeElection \rightarrow pub.srl.Bag \rightarrow bagempty \rightarrow STOP$$

$$B\_BOARD \mathrel{\widehat{=}} openElection \rightarrow \|_{serials} BOARD(serials)$$

Counter Process. The other important system process is *COUNTERS*. This works as an election authority, which counts the votes that are published on the bulletin board. The process keeps record of *place* events for each candidate. When all of the *place* events have occurred, it performs a *bagempty* event on which all *COUNTERS* processes synchronise. With the *total* event the number of total votes for each candidate is published.

$$COUNTER(cand, r) \mathrel{\widehat{=}} place?id?(i, j) \rightarrow COUNTER(cand, r+1) \\ \Box\ bagempty \rightarrow total!cand!r \rightarrow STOP$$

$$COUNTERS \mathrel{\widehat{=}} \|_{candidates} COUNTER(cand, 0)$$

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

$$SYSTEM \mathrel{\widehat{=}} VOTERS \parallel AUTHORITY \parallel BOOTH \parallel B\_BOARD \parallel COUNTERS$$

3 Automated Anonymity Verification

Our analysis of ThreeBallot uses the formal anonymity definition given in [11]. The definition of anonymity for the voting systems, also called weak anonymity, is based on observational equivalence and expressed as follows:

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

$$P[\![^{c.x,\, d.x}/_{d.x,\, c.x} \mid x \in T]\!] \equiv_{\mathrm{T}} P \tag{1}$$

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 from the original process, P, from an observer's point of view, then the process provides anonymity.

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 for ThreeBallot CSP model ($SYSTEM$) is checked by the trace equivalence:

$$SYSTEM[\![^{choose.v.1.c.x,\, choose.v.2.c.x}/_{choose.v.2.c.x,\, choose.v.1.c.x}]\!] \equiv_{\mathrm{T}} SYSTEM$$

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.

$$ABS\_SYS \mathrel{\widehat{=}} SYSTEM \setminus \{|\, alloc, place \,|\}$$

As can be seen above, the normal system is ABS_SYS, and the system where we swap two votes is $SPEC$. Therefore, if the two systems are observationally equivalent then the system provides anonymity.

$$SPEC \mathrel{\widehat{=}} ABS\_SYS[\![^{choose.v.1.c.x,\, choose.v.2.c.x}/_{choose.v.2.c.x,\, choose.v.1.c.x}]\!]$$

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 refinement $SPEC \equiv_{\mathrm{T}} ABS\_SYS$ 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 which violate anonymity. 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 Table 2, 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.

Table 2. Voting scenario 1.

Receipts		**Mini-ballots on BB**			
●	○	○	●	●	○
●	●	○	○	●	○
s_2	s_3	s_0	s_1	s_4	s_5

Table 3. Reconstruction attack 1.

choose.v.1.c.1			**choose.v.2.c.2**		
●	●	○	○	○	●
●	○	○	●	○	●
s_2	s_1	s_5	s_3	s_0	s_4

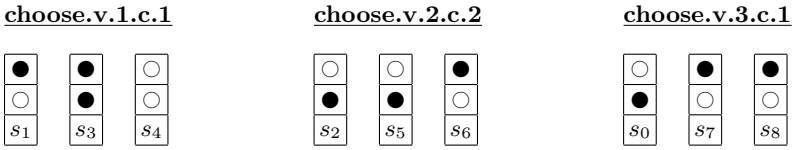
With the public information shown on the bulletin board and the receipts that the voters share with the coercer, the only way of reconstructing these votes is illustrated in Table 3. The mini-ballots s_0 and s_5 can be swapped. However, it does not affect the way the voters have voted.

Example 2. In an election with three voters and two candidates, as depicted in Table 4, 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. Table 5 shows the only possible reconstruction.

Table 4. Example 2. voting scenario

Receipts			**Mini-ballots on the BB**					
●	○	○	●	○	○	●	●	●
○	●	●	●	○	●	○	○	○
s_1	s_2	s_0	s_3	s_4	s_5	s_6	s_7	s_8

Table 5. 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 Table 6, receipt s_0 has two possible completions: it could be combined with s_2 and s_4 or s_8 (as depicted in Table 7), or with s_5 and s_7. But in either case, it represents a vote for the third candidate.

Table 6. An example voting scenario: all possible mini-ballots appear on the bulletin board

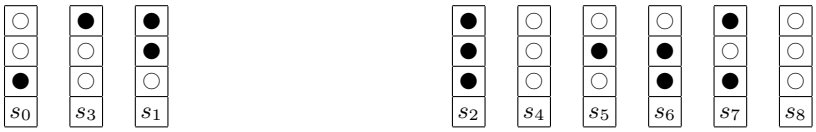
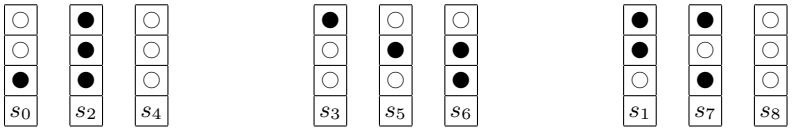


Table 7. Reconstruction attack



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

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 n 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 n 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 n copies of each such multi-ballot to the board. This means that every possible mini-ballot now appears at least n times.

However, any voter taking a singleton as a receipt will have no 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 n 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 cast during the election; $R \subset M$ is the set of all 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 serial number, 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. The SBA-pro requires at least $n \cdot r$ multi-ballots of the right type to have been cast to protect anonymity. By contrast, the SBA-multi requires at least $n \cdot 3^n$ other appropriate multi-ballots. As long as r is small, the SBA-pro is much less demanding compared with the SBA-multi. For instance, in an election with 10 candidates, the SBA-multi needs at least 590,490 multi-ballots. Unless the adversary has seen somewhere in the order of 59,000 receipts, the SBA-pro is much more likely to be satisfied.

This efficiency argument is 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. 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.

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 Conclusion

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

In our analysis, we have used an abstracted CSP model of ThreeBallot, which is 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 does not provide anonymity under various formulations of the short ballot assumption. We have in addition given a reasonable and plausible interpretation of the short ballot assumption that does in fact prevent reconstruction attacks.

Finally, 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. Table 8 illustrates the ThreeBallot verification times ("−" means no result is produced in a reasonable time).

Table 8. FDR verification times for ThreeBallot versions

	Original		**No mini-ballot empty**		**All mini-ballots appear**	
	States	Time	States	Time	States	Time
2 vtrs 2 cnds	239,905	$7.8'$	56,841	$5.3'$	240,055	$7.0'$
2 vtrs 3 cnds	4,139,347	$1''41.8'$	1,435,926	$38.3'$	4,165,428	$1''40.1'$
3 vtrs 2 cnds	−	−	67,409,391	$22''49.3'$	−	−

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

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

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

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

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

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

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

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