The Effectiveness of Receipt-Based Attacks on ThreeBallot

Kevin Henry[*], Douglas R. Stinson[†], Jiayuan Sui

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, N2L 3G1, Canada
{k2henry, dstinson, jsui}@uwaterloo.ca

Abstract

The ThreeBallot voting system is an end-to-end (E2E) voter-verifiable voting system. Each voter fills out three ballots according to a few simple rules and takes a copy of one of them home as a receipt for verification purposes. All ballots are posted on a public bulletin board so that any voter may verify the result. In this paper we investigate the effectiveness of attacks using the voter's receipt and the bulletin board. We determine thresholds for when the voter's vote can be reconstructed from a receipt, and when a coercer can effectively verify if a voter followed instructions by looking for prespecified patterns on the bulletin board. Combining these two results allows us to determine safe ballot sizes that resist known attacks. We also generalize a previous observation that an individual receipt can leak information about a voter's choices.

1 Introduction

The ThreeBallot voting system was introduced by Rivest [6] as an end-to-end (E2E) voter-verifiable election system that does not rely on any cryptography to achieve privacy. As the name implies, in the ThreeBallot system each voter completes not one, but three separate ballots (called a multi-ballot) in such a way that no single ballot should reveal information about the vote, but all three together allow the vote to be counted correctly. All completed ballots are posted on a bulletin board and the voter is allowed to take home a copy one of

[*] Supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a CGS Scholarship

[†] Supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the grant NSERC-RGPIN #203114-06

the three receipts to verify her vote was counted correctly. An overview of the system is given later in this section.

Since its introduction, ThreeBallot has been the subject of some criticisms. It was quickly pointed out by Strauss [8] that in many realistic settings the exponentially many ways a multi-ballot may be filled out, combined with the constraints imposed on a properly formed multi-ballot, make it possible to reconstruct the original multi-ballot from a voter's receipt. The exponential number of possible ballots can also be exploited by a coercer who realizes that, for large ballot sizes, the possibility of a fixed pattern of ballots occurring is small. Others have pointed out that a single receipt may also leak information about a voter's choices [3] without any reliance on the bulletin board.

In this paper we generalize previous results on so-called "leaky-receipts", as well as provide a theoretical analysis of the effectiveness of vote reconstruction and pattern requesting attacks. In the case of reconstruction and pattern attacks, we provide simulated results that back up our theoretical predictions. The most recent ThreeBallot proposal introduces the short ballot assumption (SBA) and calls for the process of "debundling", or breaking a ballot into smaller pieces to reduce the effectiveness of known attacks. If the SBA holds, then known attacks should not be practical. Using our results, we can compute definite cut-off points where the SBA holds against reconstruction and pattern attacks. Our results focus on two-candidate races, as many ballots contain a large number of yes/no issues.

1.1 Overview of ThreeBallot

A multi-ballot consists of three individual ballots, each with a unique random ID number at the bottom. If the individual ballots are attached to each other, there is a perforated edge which allows them to be separated before being placed in the ballot box. Below is a sample multi-ballot with three candidates. We omit the candidate names on the second and third ballots; however, they would appear on all three in practice.

Candidate A	O	O	O
Candidate B	O	O	O
Candidate C	O	O	O
	937856	485620	128748

The ID numbers at the bottom of each ballot are generated independently and randomly as there should be no link between the voter or the individual ballots. If the voter wishes to cast a vote for Candidate A, then, as a first step, the voter places an X exactly once for each candidate randomly across the three ballots. Next, the voter randomly marks an additional circle for Candidate A in one of the two remaining positions. A possible multi-ballot voting for Candidate A could be:

Candidate A	X	O	X
Candidate B	O	X	O
Candidate C	O	O	X
	937856	485620	128748

The voter chooses a single ballot to take home as a receipt, and has a copy of that ballot made before placing all three separated ballots into the ballot box. Once the election is over, all ballots are placed onto a public bulletin board. If there are $3n$ ballots posted on the bulletin board, then each candidate will have $n+k$ votes on the bulletin board, where k is the actual number of votes for that candidate.

To verify that her vote was counted correctly, the voter may look up her receipt via the ID number to check that it has not been altered. Because one third of the ballots on the bulletin board have been selected as receipts, an attacker has a $\frac{2}{3}$ chance of succeeding in modifying a single ballot. Thus, the probability of success rapidly becomes negligible as the number of modified votes increases. A voter's receipt also does not state who the voter voted for, so it cannot be used to prove how she voted with absolute certainty.

1.2 Previous Work

ThreeBallot was proposed by Rivest in October 2006 [6] in a paper calling for comments and suggestions. The system was later refined with some variants (called VAV and Twin) being suggested by Rivest and Smith [7]. In the interim, several issues have been raised with ThreeBallot.

Clark, Essex, and Adams [3] considered security requirements for receipts in E2E voter-verifiable voting systems, focusing on ThreeBallot, PunchScan [4, 5], and Prêt-à-Voter [2]. They propose that:

1. A receipt should contain no information that increases the ability of a coercer to determine the voter's choices, and,
2. A receipt should not increase an adversary's chance of modifying ballots without detection.

All three systems were found to satisfy the second property; however, ThreeBallot failed to satisfy the first. Section 2.1 contains more details and a generalization of their findings.

Strauss [8] and Appel [1] have each pointed out usability flaws and potential receipt buying attacks against ThreeBallot. In addition, Strauss also showed that, in many settings, it is possible to reconstruct a vote from a single receipt using the bulletin board [9]. We refer to this as a reconstruction attack and present Strauss' results alongside our own in Section 3.

2 Two-Candidate Races

Two-candidate races are of particular interest. Not only do many real world elections contain several two-candidate races, but some attacks, such as the two-candidate attack presented in Section 2.1, are most effective when applied to two-candidate races. As the results in Section 3 and 4 rely on the probabilities of certain receipts occurring, we now present a basic analysis of two-candidate races.

The following table demonstrates the eighteen different ways a two-candidate multi-ballot can be completed.

A	X X O	X O X	X X O	A	X O O	X O O	O X O
B	X O O	X O O	O X O	B	X X O	X O X	X X O
A	X O X	O X X	O X X	A	O O X	O X O	O O X
B	O O X	O X O	O O X	B	X O X	O X X	O X X
A	X X O	X O X	O X X	A	O O X	O X O	X O O
B	O O X	O X O	X O O	B	X X O	X O X	O X X

Votes for Candidate A — Votes for Candidate B

In general, the number of ways a multi-ballot with r races, where race i has c_i candidates, can be filled out is given by

$$\prod_{i=1}^{r} \left(3^{c_i} c_i\right).$$

The eighteen possible valid two-candidate multi-ballots yield 54 possible receipts a voter may choose from. Of these receipts, 12 of them contain two votes, 12 of them contain no votes, and the remaining 30 contain exactly one vote. If we assume the voter fills out her multi-ballot and chooses her receipt randomly, then the probability of each of the four possible receipts being chosen is as follows:

$\begin{array}{|c|}\hline \text{X}\\ \text{X}\\ \hline\end{array}$	$\begin{array}{|c|}\hline \text{O}\\ \text{O}\\ \hline\end{array}$	$\begin{array}{|c|}\hline \text{X}\\ \text{O}\\ \hline\end{array}$	$\begin{array}{|c|}\hline \text{O}\\ \text{X}\\ \hline\end{array}$
$\Pr(\cdot) = \frac{2}{9}$	$\Pr(\cdot) = \frac{2}{9}$	$\Pr(\cdot) = \frac{5}{18}$	$\Pr(\cdot) = \frac{5}{18}$

2.1 Generalized Two-Candidate Attack

Consider the table of possible two-candidate multi-ballots from the previous section. There are 9 ballots voting for A, yielding 27 possible receipts. Of these 27 receipts, 12 of them are $\begin{array}{|c|}\hline \text{X}\\ \text{O}\\ \hline\end{array}$. From the 9 ballots voting for B, only 3 of them are $\boxed{\begin{matrix}\text{X}\\\text{O}\end{matrix}}$. Hence, 12 of 15, or 80% of these receipts correspond to a vote for A. Symmetrically, we can use the opposite receipt to infer a vote for B with the same confidence. We call this imbalance in receipt distribution the two-candidate attack, although the same idea may be applied to larger races, albeit with less effectiveness.

Clark et al. [3] observed this two-candidate attack as a violation of one of the desired properties of a receipt-based system, namely that a voter's receipt should not increase a coercer's ability to determine the voter's choice. Their model was limited to information that a receipt leaks by itself without knowledge of the bulletin board. Because the bulletin board presents additional information to an adversary, it can be utilized to strengthen the two-candidate attack. Instead of assuming that the voter has chosen her candidate randomly, we need only consider that the pattern used on the multi-ballot is random with respect to the voter's choice, as well as which receipt was the receipt taken.

Let X_A be the number of multi-ballots voting for A and let X_B be the number of multi-ballots voting for B. These values can easily be computed from the bulletin board. We now determine the expected number of occurrences of $\boxed{\begin{matrix}\text{X}\\\text{O}\end{matrix}}$.

Of the nine possible multi-ballots voting for A, six of them contain a single copy of $\boxed{\begin{matrix}\text{X}\\\text{O}\end{matrix}}$ and three of them contain two copies of $\boxed{\begin{matrix}\text{X}\\\text{O}\end{matrix}}$. Thus, the expected number of $\boxed{\begin{matrix}\text{X}\\\text{O}\end{matrix}}$ receipts taken by voters who chose A is

$$X_A \left(\frac{6}{9} \cdot 1 + \frac{3}{9} \cdot 2 \right) = \frac{4X_A}{3}.$$

Similarly, from the multi-ballots voting for B we have six containing no copies and three contains a single copy of $\boxed{\begin{matrix}\text{X}\\\text{O}\end{matrix}}$. The expected number of $\boxed{\begin{matrix}\text{X}\\\text{O}\end{matrix}}$ receipts taken by voters who chose B is then

$$X_B \left(\frac{6}{9} \cdot 0 + \frac{3}{9} \cdot 1 \right) = \frac{X_B}{3}.$$

Using these values, we can now solve for the probability that a voter who takes receipt $\boxed{\begin{matrix}\text{X}\\ \text{O}\end{matrix}}$ has voted for A:

$$
\begin{aligned}
\Pr\left[\text{vote for A} \mid \boxed{\begin{matrix}\text{X}\\ \text{O}\end{matrix}}\right] &= \frac{\Pr\left[\text{vote for A } \cap \boxed{\begin{matrix}\text{X}\\ \text{O}\end{matrix}}\right]}{\Pr\left[\boxed{\begin{matrix}\text{X}\\ \text{O}\end{matrix}}\right]} \\
&= \frac{\frac{4X_A}{3}}{\frac{4X_A}{3}+\frac{X_B}{3}} = \frac{4X_A}{4X_A+X_B}.
\end{aligned}
$$

We can similarly solve for the probability that a voter who takes receipt $\boxed{\begin{matrix}\text{O}\\ \text{X}\end{matrix}}$ has voted for B as

$$
\Pr\left[\text{vote for B} \mid \boxed{\begin{matrix}\text{O}\\ \text{X}\end{matrix}}\right] = \frac{4X_B}{X_A+4X_B}.
$$

Setting $X_A = X_B$ yields the expected $\frac{4}{5} = 80\%$ probability in the original two-candidate attack. Intuitively, this result is exactly as one would expect. If more people vote for candidate A, then it is more likely that the receipt $\boxed{\begin{matrix}\text{X}\\ \text{O}\end{matrix}}$ came from a vote for A.

3 Reconstruction of Ballots

Shortly after ThreeBallot was proposed, Strauss [9] observed that, given a single receipt and the bulletin board, it is possible to reconstruct the original ThreeBallot in many realistic election settings. This was accomplished through the use of simulated elections on a computer. In this section we recreate Strauss' work and provide a theoretical basis for when reconstruction is possible with high probability.

3.1 Theoretical Results

The latest revision of ThreeBallot calls for the process of "debundling", or breaking a large ballot into several smaller pieces, each posted separately, so as to minimize the effectiveness of the reconstruction attack. In this section we develop a formula that can be used to compute the maximum number of voters or the maximum number of two candidate races a multi-ballot may contain before the reconstruction attack becomes feasible. The derived formula compares favorably with simulated results presented in the next section.

A set of three ballots forms a valid triple if there are exactly two votes for one of the candidates and a single vote for the other. This can occur in one of two ways: Each ballot contains a single vote (two for the same candidate, one for the other), or each receipt contains a different number of votes.

Recall from Section 2 the probability for each individual receipt to occur. Strictly speaking, the ballots on the bulletin board are not totally independent; however, when the number of voters is large, the inter-dependency is minimal and we assume independence to make an approximation. Thus, the probability that a randomly chosen triple will be valid is estimated to be:

$$6 \cdot \left(\frac{5}{18}\right)^3 + 6 \cdot \left(\frac{2}{9} \cdot \frac{2}{9} \cdot \frac{5}{9}\right) = \frac{95}{324}.$$

Extending this result to a multi-ballot containing r two-candidate races, the probability a triple of votes is valid for all r races is estimated to be:

$$\left(\frac{95}{324}\right)^r$$

and the probability that a triple of votes is not valid for at least one of the races is estimated to be:

$$1 - \left(\frac{95}{324}\right)^r.$$

Given a receipt, we can reconstruct the original multi-ballot if there is a unique valid triple containing the given receipt. If there are n voters, then there are $\binom{3n-1}{2}$ possible pairs of votes that can form a triple with the given receipt. The probability that all but one of these pairs do not form a valid triple (at least one valid triple must exist if the original multi-ballot was constructed appropriately) is estimated by

$$\left(1 - \left(\frac{95}{324}\right)^r\right)^{\binom{3n-1}{2}-1}.$$

Using this formula, we can plot the probability of reconstruction for a fixed value of r or n.

Figure 1 is a plot of the reconstruction probability for varying values of r when n =100, 1000, 10000.

3.2 Simulated Results

Strauss' results were generated by querying a dozen random ballots over several elections to determine the probability that the original multi-ballot could be reconstructed. The query process involved matching every possible pair of ballots on the bulletin board to the query receipt, adding any compatible matches to a "query set". If the set contains only a single pair, then the multi-ballot has been reconstructed. If not, the same process is repeated recursively on each

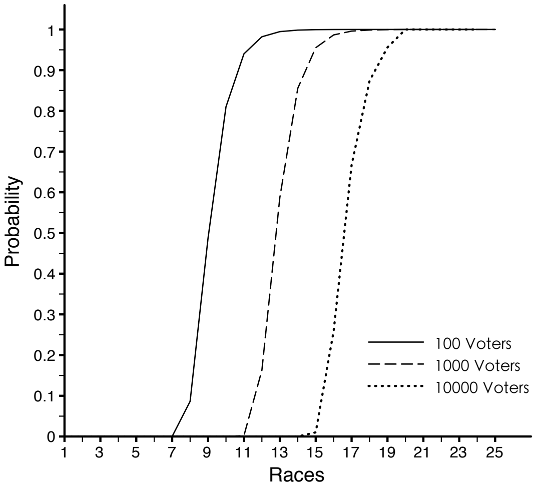

Figure 1: Probability of reconstruction success for $n = 100$, 1000, 10000.

ballot in the query set, removing any ballot from the current query that leads to a unique match. As noted by Strauss, it may be possible to find additional matches using a more complicated approach; however, this simple approach is still very effective.

To generate our results, we use two variants of Strauss' approach. The first is a simple "single-pass" approach, so named because we simply query each ballot on the bulletin board once to see if it leads to a unique query set.

Algorithm 1: Single-Pass Reconstruction

```
for each ballot $b$ on the bulletin board do
    for each remaining ballot pair $p$ on the bulletin board do
        if $b$ and $p$ form a valid triple then
            add $p$ to the query set
        end
    end
    if query set is unique then
        remove $b$ and $p$ from the bulletin board
        add $b$ and $p$ to the set of reconstructed ballots
    end
end
```

A more effective version of this is the "multi-pass" approach. Each time a ballot is reconstructed and removed from the bulletin board, it is possible that we have now made the query set for some previously queried ballot unique. Thus, we simply re-run the algorithm repeatedly until no new ballots can be reconstructed. This is similar to Strauss' recursive queries.

Algorithm 2: Multi-Pass Reconstruction

```
run Algorithm 1
while at least one new multi-ballot was reconstructed do
    run Algorithm 1 again
end
```

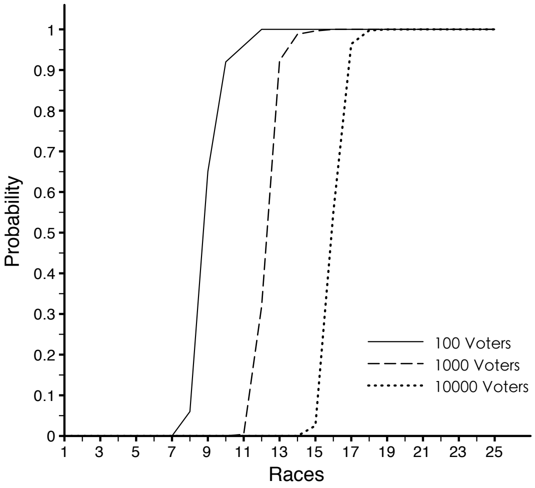

Figure 2: Effectiveness of the single-pass approach on two-candidate races for $n = 100$, 1000, 10000.

Figure 2 demonstrates the results of the single-pass algorithm for 100, 1000, 10000 voters on varying numbers of two-candidate races. The simulated elections assumed that each multi-ballot was filled out randomly, but correctly, by the voter. Figure 3 shows corresponding results for the multi-pass approach. The two graphs are similar; however, the single-pass approach grows from near 0% to 90% over 2-3 races, while the multi-pass approach grows over just a single race. This suggests that once a small amount of multi-ballots can be reconstructed after a single pass, we will be able to reconstruct most of the multi-ballots with subsequent passes. The simulated single-pass approach is very similar to the theoretical prediction from the previous section, as can be seen by comparing Figures 1 and 2.

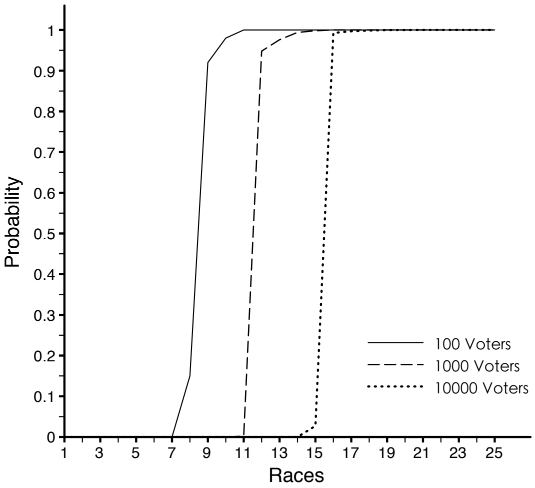

Figure 3: Effectiveness of the multi-pass approach on two-candidate races for $n = 100$, 1000, 10000.

Our results differ from the results of Strauss, who found that the transition from 0% to 90% success took place at 11, 17, and 23 races for 100, 1000, and 10000 voters respectively. Our simulation and theoretical predictions suggest that reconstruction begins to become effective at 8, 12, and 15 races respectively. A possible explanation for these differences could be the use of different simulation techniques. Strauss randomly queried a dozen ballots for each of 30 elections, using recursive queries when a unique match was not found, whereas we queried all $3n$ possible receipts for each value of n, using multiple passes until no new matches were found.

Our theoretical analysis is very similar to the single-pass approach; however, the theoretical approach does not account for ballots that have been reconstructed and removed from the bulletin board in earlier queries. Calculating our theoretical prediction from the single-pass rather than multi-pass approach is not an issue when attempting to determine safe ballot sizes, as both approaches become effective at the same point and only differ by the rate at which they grow more effective. Thus, our theoretical analysis of the single-pass approach is still useful for determining safe ballot sizes.

4 The ThreePattern Attack

Recall that, as the number of candidates on a multi-ballot increases, the number of ways the multi-ballot can be filled out increases exponentially. Thus, for larger ballot sizes, it is possible that the number of ways to fill out a multi-ballot may be far greater than the number of voters. An attacker can exploit this fact by offering payment to a voter if a given set of three receipts appears on the bulletin board. Because the requested pattern may only occur with small probability, the attacker can be reasonably certain that the coerced voter properly followed instructions if the requested pattern can be found on the bulletin board. We refer to this attack as the ThreePattern attack.

The goal of this section is to determine when the ThreePattern attack is ineffective. We will call the ThreePattern attack ineffective if the chance of any given pattern occurring is greater than some threshold, say 99%. Our analysis is independent of this threshold and allows election officials to choose whichever value they deem necessary. As in previous sections, we focus on two candidate races and assume that each multi-ballot is constructed randomly, but correctly.

4.1 Theoretical Results

Assume the attacker has chosen a pattern to use for the ThreePattern attack. Let p_i, $i = 1, 2, 3$, be the probability that ballot i of the requested pattern occurs on the bulletin board. If each ballot contains r races, then $p_i = \prod_{j=1}^{r} p_{i_j}$, where p_{i_j} is the probability that race j on receipt i matches the requested pattern. Finally, let X be a random variable denoting the number of times that the requested pattern occurs on the bulletin board. We now calculate $\Pr[X \geq 1] \geq 0.99$, i.e., the probability that a given pattern occurs at least once with probability greater than or equal to 99%.

The probability that none of the $3n$ ballots on the bulletin board match the requested ballot i is $(1 - p_i)^{3n}$. Thus, the probability that at least one of the ballots on the bulletin board matches is given by $1 - (1 - p_i)^{3n}$, and the probability that all three ballots occur is given by

$$\Pr[X \geq 1] = \prod_{i=1}^{3} \left(1 - (1 - p_i)^{3n}\right) = \prod_{i=1}^{3} \left(1 - \left(1 - \prod_{j=1}^{r} p_{i_j}\right)^{3n}\right).$$

As in our analysis of the reconstruction attack, we assume that individual ballots are independent to make an approximation.

Figure 4 shows a plot of this formula for a specific multi-ballot pattern with a varying number of races for $n = 100$, 1000, 10000.

With just 100 voters, the ThreePattern attack is feasible if the ballot size grows larger than two races, although the probability of the requested pattern

X X	X O	O O
X X	X O	O O
⋮	⋮	⋮
X X	X O	O O

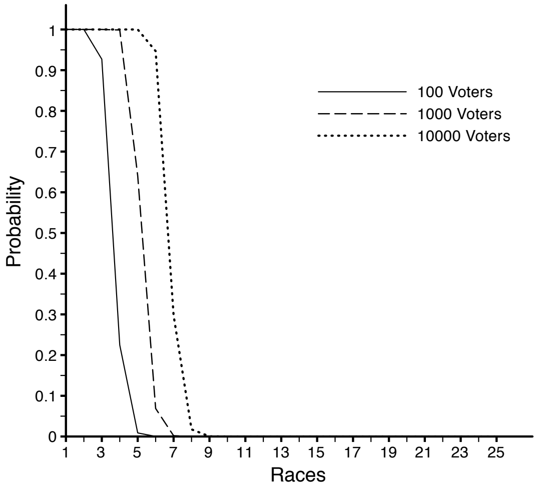

Figure 4: Probability that the given pattern occurs at least once on the bulletin board.

appearing is still 80% for three races. Recall from the previous section that the Reconstruction attack for $n = 100$ begins to become effective at seven races, about the same point where the ThreePattern attack becomes effective for $n = 10000$. This suggests that the effectiveness of the ThreePattern attack should be used as a guideline when determining when the short ballot assumption is satisfied; however, this creates impractical limitations on ballot sizes, especially when the number of voters is small.

We now turn our attention to the probability that a given pattern occurs at least m times. As this situation is less likely than a pattern occurring just once, and a coercer may wish to coerce $m > 1$ voters at a time, a coercer may instruct several voters to vote using the same pattern, offering payment if and only if at least m copies of the pattern appear on the bulletin board. The probability of at least m copies of the requested ballot i occurring is given by

$$\begin{aligned}\Pr[X \geq m] &= 1 - \Pr[X = 0] - \Pr[X = 1] - \ldots - \Pr[X = m - 1] \\ &= 1 - \sum_{k=0}^{m-1} \left(\binom{3n}{k} (1 - p_i)^{3n-k} p_i^k \right) \\ &= 1 - \sum_{k=0}^{m-1} \left(\binom{3n}{k} \left(1 - \prod_{j=1}^{r} p_{i_j} \right)^{3n-k} \left(\prod_{j=1}^{r} p_{i_j} \right)^{k} \right).\end{aligned}$$

Thus, the probability of all three requested ballots occurring at least m times is given by

$$\Pr[X \geq m] = \prod_{i=1}^{3} \left[1 - \sum_{k=0}^{m-1} \left(\binom{3n}{k} \left(1 - \prod_{j=1}^{r} p_{i_j} \right)^{3n-k} \left(\prod_{j=1}^{r} p_{i_j} \right)^{k} \right) \right] .$$

Figure 5 shows a plot of this formula for $m = 5$ with varying numbers of races, using the same pattern specified earlier. The plot is similar to Figure 4, however the probability begins to drop around one race earlier, and the effectiveness grows slightly faster.

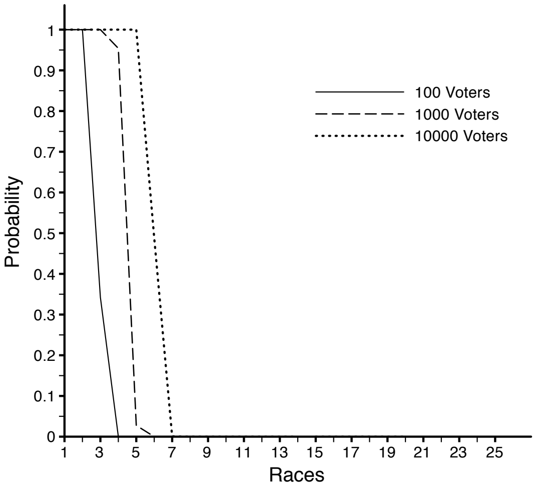

Figure 5: The probability that a specific pattern occurs at least 5 times.

As a final consideration, we examine the case where a coercer may request m different patterns from m voters. In light of the result for $\Pr[X \geq 1]$, we can simply take the product of this formula for each of the m patterns,

$$\prod_{i=1}^{m} \prod_{j=1}^{3} \left(1 - (1 - (p_j)_i)^{3n} \right) ,$$

which allows us to solve for the number of voters required to render the Three-Pattern attack ineffective.

It is interesting to note that it is more effective for the coercer to request m copies of the same pattern, rather than requesting m different patterns. However, if a disproportionate number of ballots are repeated on the bulletin board, it may catch the attention of election officials. This means the coercer must decide between the more effective but easier-to-detect form of coercion, or the less effective but harder-to-detect form.

4.2 Simulated Results

To verify the effectiveness of the ThreePattern attack we simulated a number of different elections for varying numbers of voters and races, and tested how many times a pre-specified pattern occurred. Algorithm 3 details the method used. Given a set of m different patterns and the corresponding number of occurrences, we simply conduct a random election and verify whether or not each of the patterns occurred sufficiently many times.

```
Algorithm 3: ThreePattern Lookup
Input: The number of trials $n$, patterns $p_1, \ldots, p_m$, and the requested
       number of occurrences $k_1, \ldots, k_m$ of each pattern
$numSuccess \leftarrow 0$
for $i = 1 \ldots n$ do
    generate a random election outcome
    for $j = 1 \ldots m$ do
        $c \leftarrow$ the number of occurrences of pattern $p_j$
        if $c >= k_j$ then
            $success_j \leftarrow true$
        else
            $success_j \leftarrow false$
        end
    end
    if $success_j = true\ for\ j = 1 \ldots m$ then
        $numSuccess \leftarrow numSuccess + 1$
    end
end
output $numSuccess/n$
```

Figure 6 demonstrates the output of Algorithm 3 for varying numbers of races over 100 trials using a random requested pattern for each election. The two plots are nearly identical to their corresponding theoretical predictions given in Figures 4 and 5.

Some might argue that requiring a 99% chance of any given pattern occuring is an unnecessarily strong requirement. In practice, we might be satisfied with any threshold over 50%, meaning that a coercer's chance of success is less than 50%. However, by examining both the theoretical and simulated plots, we see that lowering the threshold to 50% will allow ballots to grow by only a single race in most cases, if at all. This is due to the exponential growth in possible ballot configurations as more races are added. For each case of n = 100, 1000,

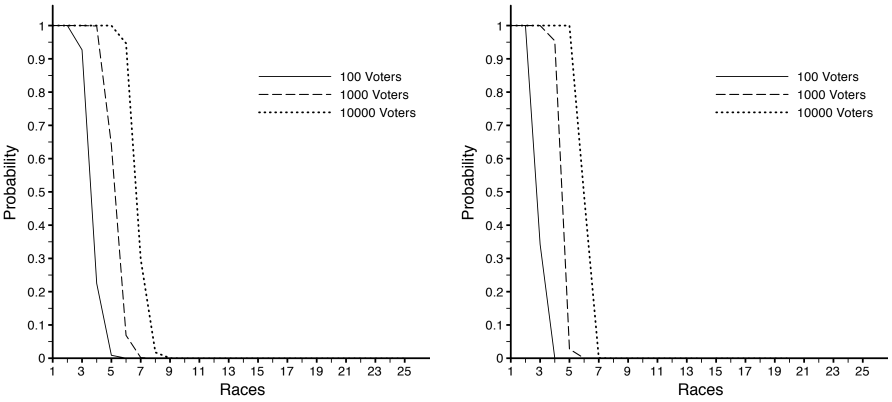

Figure 6: 100 trials of the ThreePattern attack for 100, 1000, and 10000 voters over varying numbers of races. The left graph shows $k = 1$, the probability of a single instance occurring, while the right graph shows $k = 5$, the probability that the pattern occurs 5 times.

10000, there is approximately a three-race window in which the probability of a given pattern occuring transitions from 100% to near 0%.

5 Concluding Remarks

We have presented a detailed analysis of known receipt-based attacks against the ThreeBallot voting system, focusing on two-candidate races. Our generalization of the two-candidate attack allows an adversary to take advantage of the bulletin board to increase the probability of determining a voter's vote, given their receipt. In the case of reconstruction and pattern attacks, we determined formulas that can be used to compute the number of races a multi-ballot may contain before either type of attack may apply. The following table summarizes the maximum safe ballot size for 100, 1000, and 10000 voters.

Voters	Reconstruction	ThreePattern ($k = 1$)	ThreePattern ($k = 5$)
100	7	2	2
1000	11	4	3
10000	15	6	5

It appears that the ThreePattern attack is more of a concern than the reconstruction attack, as it becomes effective far earlier. Election officials must determine which value of k (the number of repeated patterns a single coercer may ask for) they wish to plan for in a given election. This will allow them to compute the maximum ballot size that satisfies the short ballot assumption. Ballots can then be "debundled" into appropriately sized sub-ballots that resist known attacks. Unfortunately, the required ballot sizes to resist the Three-Pattern attack are relatively small, and may limit the use of ThreeBallot in situations where debundling into small enough ballots is not possible.

References

[1] A. Appel. How to defeat Rivest's ThreeBallot Voting System., Princeton University, `http://www.cs.princeton.edu/~appel/papers/DefeatingThreeBallot.pdf`, October, 2006.

[2] D. Chaum, P. Ryan, S. Schneider. A Practical Voter-Verifiable Election Scheme, Technical Report of University of Newcastle, CS-TR:880, 2005.

[3] J. Clark, A. Essex, C. Adams. On the Security of Ballot Receipts in E2E Voting Systems, *Proceedings of Workshop On Trustworthy Elections (WOTE), 2007.*

[4] K. Fisher, R. Carback, A. Sherman. Punchscan: Introduction and System Definition of a High-Integrity Election System, *Proceedings of Workshop on Trustworthy Elections (WOTE), 2006.*

[5] S. Popoveniuc, B. Hosp. An Introduction to Punchscan, *Proceedings of Workshop on Trustworthy Elections (WOTE), 2006.*

[6] R. Rivest. The ThreeBallot Voting System, `http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf`, October 2006.

[7] R. Rivest, W. Smith. Three Voting Protocols: ThreeBallot, VAV, and Twin, *Proceedings of USENIX/ACCURATE Electronic Voting Technology Workshop (EVT), 2007.*

[8] C. Strauss. The trouble with Triples: A critical review of the triple ballot (3ballot) scheme. Part 1, Verified Voting New Mexico, `http://www.cs.princeton.edu/~appel/voting/Strauss-TroubleWithTriples.pdf`, October, 2006.

[9] C. Strauss. A Critical Review of the Triple Ballot Voting System. Part2: Cracking the Triple Ballot Encryption, Draft Version 1.5, Verified Voting New Mexico, `http://www.cs.princeton.edu/~appel/voting/Strauss-ThreeBallotCritique2v1.5.pdf`, October, 2006.