Plinko: Polling with a Physical Implementation of a Noisy Channel

Chris Alexander
University of Waterloo
Waterloo, ON, Canada
cialexan@cs.uwaterloo.ca

Joel Reardon*
Google Switzerland GmbH
Zürich, Switzerland
reardon@google.com

Ian Goldberg
University of Waterloo
Waterloo, ON, Canada
iang@cs.uwaterloo.ca

Abstract

We give a practical polling protocol that is immune to tampering by either the pollster or the responder. It preserves responders' privacy in the manner of Warner's Randomized Response Technique, is easily understood without any knowledge of cryptography, and does not require the use of computers or other electronics. The key is to use physical noisy channels commonly found in lottery or game-show settings, which can deliver the desired properties without relying on a mechanism which is unfamiliar to the responder.

1 Introduction

Election systems have been well studied; there exists a wide variety of protocols to ensure accurate and private elections under various circumstances and assumptions. Many of these protocols [1, 3, 8] prioritize usability, and great strides have been made in systems that are both secure enough to be trusted, and simple enough that most voters may vote correctly with minimal instruction.

However, the related problem of polling has received much less attention. As in voting, polling requires that all responses have equal weighting, that the final tally be accurate, and that each individual response be kept private. However, there are also important differences: the incorrect recording of an individual response is permissable, so long as the final tally can provide statistically useful polling data. Moreover, the informal nature of many polls precludes trusted third parties or scrutineers. These differences make polling schemes both simple to implement and difficult to verify.

While there are well-known methods for preserving the privacy of poll responses, even in the absence of a trusted third party [10], such methods traditionally either allow cheating by one or more participants [10], or rely on mechanisms that are difficult for a layperson to understand and verify [2]. More recently, some promising work by Moran and Naor [6] attempted to provide simple and secure polling protocols, but the protocols only worked for very specific parameter choices; the generalized forms are too cumbersome for ordinary polling use. In this paper we present a polling method which acheives all of Moran and Naor's desired properties, while remaining easily generalized to a wider range of practical parameter values.

1.1 The polling problem

Poll questions span many topics, but for those questions that are sensitive in nature—for example, dealing with criminal, political, or medical issues—it is particularly difficult to obtain correct results. In such a poll, the responder may feel embarrassed by her true answer and choose instead to lie, thus distorting the poll. This leads to undetected systemic measurement error regardless of the sample size. To prevent this, pollsters attempt to provide some assurance of privacy. One simple method is an anonymous survey: answers are submitted without identifiers, and their

[*] Work done while a student at the University of Waterloo

order is permuted before they are examined. However, it has been shown that anonymous responses are still prone to underreporting of stigmatized groups or behaviours [9]. A better method is Warner's Randomized Response Technique (RRT) [10]. Here, a pollster wishes to learn if a responder belongs to a stigmatizing group A, or the rest of the population $\bar{A}$. The responder is given a spinner which points to A with probability p and $\bar{A}$ with probability $(1-p)$. Without showing the spinner result to the pollster, the responder then answers the question: "Do you belong to the group indicated on the spinner?" Since p is known, it can be used to estimate π, the true proportion of responders in A. At the same time, it is clear that no matter what answer is given, the pollster cannot be certain as to whether or not a particular responder belongs to A.

As p approaches 0.5, better privacy is afforded; however, it comes at the cost of less useful data. Consequently, a larger sample size and greater polling cost is required to accurately estimate π. When $p = 0.5$ we have a random channel whose outputs have no statistical correlation to its inputs. Conversely, values of p near 1, by not providing sufficient privacy to the responders, lead some of them to falsify their responses. The goal is to use a value of p which is low enough to encourage honest responses, yet high enough to bound the confidence interval for π without undue cost. Many later variations of RRT have attempted to reduce the variance while preserving the privacy aspect [4, 5]. Unfortunately, reducing the variance often comes at the expense of usability; extra rounds, asymmetry and complicated calculations make the protocols harder to explain and use. As the desired value of p will depend on the particulars of the polling situation, we aim to provide a simple mechanism which can be tuned to a wide variety of p values.

In general, schemes based on RRT effectively prevent the pollster from cheating. Ideally, the pollster would be unable to learn the true value of any response with probability higher than p. This should be true even if she deviates from the protocol, aborts the protocol after a certain step, or lies about the security properties her system guarantees. In keeping with [6], we say that protocols acheiving this goal are *pollster immune*. In a pollster-immune protocol, the only protocol properties that we expect a responder to recognize are those that can be verified without any specialized training or the use of a third party. Since it is trivial for a pollster to mislead a layperson about the security properties present in a piece of software, or the hardware on which it is running, we do not consider protocols involving computers to be pollster immune.

The attacks that RRT schemes generally fail to prevent are those launched by the responder. Here, the responder attempts to corrupt the poll results to suit her own ends. For example, if a political poll is believed to have an impact on an upcoming election, responders may deviate from the protocol in an attempt to indicate their preferred party with a probability higher than p, thus skewing the results in that party's favour. In Warner's RRT, the responder could simply ignore the spinner result and answer as if the higher probability question had been asked. In other schemes, such as the scratch-off cards in [6], it may be preferrable to observe the source of randomness and abort the protocol if the random value generated is not to the responder's liking. If a protocol defends against all such attacks, we say that it is *responder immune*.

Although many polling protocols exist, we are not aware of any that satisfy all of our desired properties: having a level of perceived privacy comparable to RRT, having understandable instructions, being pollster immune, being responder immune, and allowing a responder to easily and independently verify each of these claims. Moran and Naor developed two good attempts [6], but neither is simultaneously pollster and responder immune, nor are they easy to adjust to achieve an arbitrary value of p.

1.2 Our results

In this paper, we give a usable, understandable, pollster- and responder-immune RRT protocol that can be generalized to a wide range of p values. We explain how it satisifies each property in turn, and how these properties can be verified intuitively by a responder with little knowledge of probability theory, as well as verified rigorously by an auditor. The key is to use a physical noisy channel whose randomness is not controlled by either party. In particular, we borrow the Plinko board from a popular American game show ("The Price is Right")—a random channel that is more familiar and intuitive to responders than any cryptographic protocol we have seen.

2 Plinko Board Design

The first step in this protocol is to design and create a Plinko board that comes as close as possible to acheiving the desired value of p. In this section we describe how to design such a board.

For our purposes, a Plinko board is a symmetric arrangement of rows of pegs. The pegs are spaced such that a chip passing between any two adjacent pegs will land on a third peg and bounce in either direction with $P[\text{left}] = P[\text{right}] = 0.5$. All openings at the top of the board are sealed except for two spaced equally about the centre. These slots correspond to positive and negative responses that the responder can offer for the polling question. Openings exist at the bottom of the board which allow chips to fall into two or more labelled boxes; see Figure 1 for an example of the layout.

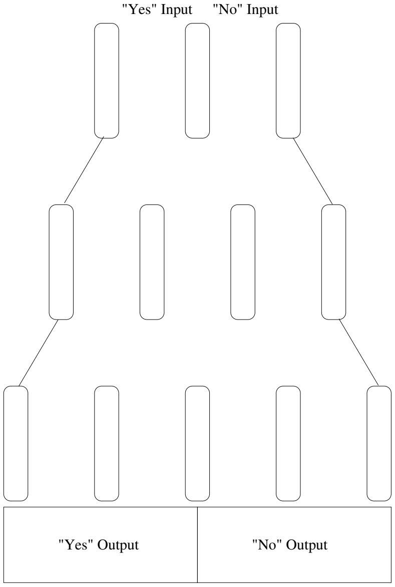


Figure 1: Layout of a Plinko board with $h = 2$, $w = 4$, and $d = 1$.

In this discussion, let:

- h be the height of the board, measured by the number of pegs that a chip will hit on a single pass through the machine;
- w be the width of the board, measured by the number of open slots at the bottom of the machine.

First, observe that w should be even; otherwise there would exist a slot at the bottom of the board perfectly centered between the two inputs. Since the board is symmetric, the probability of a chip reaching that slot is the same regardless of which input slot it was dropped in. As such, a chip arriving in this bottom slot reveals no information at all about π. A similar board with either one row more or one less would have no bottom slots with this property, and as such would be both more efficient and more closely match the original RRT.

Next, we discuss the number of boxes that should be placed below the output slots. In theory, a different box could be used for each output slot. However, this would alter the equivalent value of p in the RRT scheme randomly; a chip that bounces into the rightmost output slot is much more likely to come from the rightmost input than the leftmost (p close to 1). On the other hand, a chip in a more central output slot could easily have come from either input (p close to 0.5). This may be desirable in some applications, but for our purposes we use only two boxes: one for all outputs to the left of the board's center, and one for all outputs to the right. We can still estimate π based on the number of chips appearing in each box, but now every observed chip appears to have the same probability p of emerging from the machine on the same side as it was inserted. Using only two boxes simplifies the analysis, more closely resembles the original RRT, and provides better privacy if the chip exits on the edge of the board.

With that in mind, we define a third parameter of the board. Let d be the distance between the two inputs, measured in slots. (Adjacent slots have $d = 1$.) Note that in order to avoid having an output slot in the centre of the board, $h + d$ must be odd. The following property can easily be seen to hold: if a chip in the left input bounces right $\frac{h+d+1}{2}$ times, it will end up in an output slot on the right half of the board; if it bounces right $\frac{h+d-1}{2}$ times, it will end up in an output slot on the left half of the board. Of course, if $d > h$, then the chip cannot switch sides, so we require that $d < h$ as well.

		Height of Plinko Board (h)											
		2	3	4	5	6	7	8	9	10	11	12	13
d	1	0.750	x	0.688	x	0.656	x	0.637	x	0.623	x	0.613	x
	2	x	0.875	x	0.812	x	0.773	x	0.746	x	0.726	x	0.709
	3	x	x	0.938	x	0.891	x	0.855	x	0.828	x	0.806	x
	4	x	x	x	0.969	x	0.938	x	0.910	x	0.887	x	0.867
	5	x	x	x	x	0.984	x	0.965	x	0.945	x	0.927	x
	6	x	x	x	x	x	0.992	x	0.980	x	0.967	x	0.954

Table 1: Probability p for Plinko boards with small parameter values. An “x” indicates a configuration that does not satisfy the requirements that $d + h$ is odd and that $d < h$.

With this terminology, we now proceed to board design. Given a target value p, our goal is to find a (d, h) pair such that $P[\text{left output}|\text{left input}] = p$. We assume that w is large enough that any chips striking the left side of the board and bouncing back towards the center cannot then end up in the right output box, except with insignificant probability. Thus, this problem is equivalent to finding a pair (d, h) such that flipping h fair coins will yield $\frac{h+d+1}{2}$ or more heads with probability p. This has a well-known solution: approximate the binomial distribution of coin flips with the normal distribution. For our purposes, we may use either premade charts or mathematical software to find a value z such that $P[\text{measurement on normal dist} \leq z] = p$. Then, we scale the result for our Plinko board. If x denotes the mean number of heads, and σ denotes the standard deviation, then:

$$z = \frac{\frac{h+d+1}{2} - \frac{1}{2} - x}{\sigma} = \frac{\frac{h+d+1}{2} - \frac{1}{2} - \frac{h}{2}}{\sqrt{\frac{h}{4}}} = \frac{d}{\sqrt{h}}$$

Any near-solution to this equation with integer values for d and h (of opposite parity) will yield a Plinko board with the desired properties. We acknowledge a tradeoff here: in general, creating a board realizing the desired p to a high degree of accuracy requires a large h value. If this is not feasible, then a less accurate, smaller board is possible. In particular, the boards described in Table 1 represent the easiest p values to achieve. Regardless of board height, w should be chosen such that any chip striking an outer wall falls into that side's output bin with high probability.

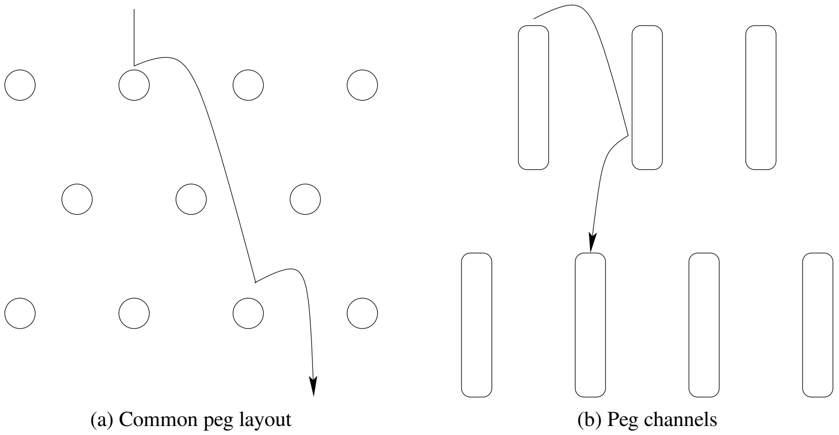
It should be noted that the normal approximation is very accurate for $h \geq 10$. For small values, the binomial distribution can be used directly. Table 1 shows several small combinations of d and h, and the resulting p values.

When actually constructing the board, care should be taken to make sure that the construction matches our mathematical assumptions. For example, while many Plinko boards consist of a simple array of cylindrical pegs, this setup allows the effect of one bounce to have a large impact on the next bounce. In our model, however, we assumed that each event of a chip striking a peg was independent of all others. Figure 2 shows how to reduce the effect of lateral movement before each peg strike: we force the chip to travel down small channels between peg strikes.

Finally, we note that once the parameters have been determined, the actual board construction is quite inexpensive. We created a sample board with $p = 0.773$ for less than $100 CDN, including the acrylic cover and a large stack of chips; a photo of our Plinko board can be seen in Figure 3. There are also companies specializing in custom-built Plinko boards for higher-cost, lower-effort construction [7].

3 Security Properties

Assuming that a board has been constructed with a known value of p, we next discuss its proper use. The output boxes are locked and firmly attached to the board. A rigid, transparent cover is attached to the front of the machine; plexiglass or acrylic are appropriate materials for the cover. There should be a large pile of chips which are not easily distinguishable. For each responder, the pollster first explains the protocol to her, along with simple explanations of the privacy it affords. Next, he gives her an even number of chips and a marker. She may then mark a single chip as the “true” chip; all others become “test” chips instead. She then proceeds to the Pinko board and drops the chips into the machine one at a time. The marked side of the true chip should face inward, so anyone watching the board cannot distinguish the true chip from the test ones. We let b be the true answer of the responder: 1 if placed in the left input, 0 if placed in the right input. After several runs of the protocol, the pollster opens the output boxes and counts the


(a) Common peg layout

(b) Peg channels

Figure 2: Decreasing the magnitude of the effect of one bounce on the next.

number of marked chips in each. Let us call the total number of marked chips n, and the total in the left output box n_1.

We claim that this protocol is both pollster and responder immune. First we first consider a dishonest responder.

3.1 Responder Immunity

A dishonest responder attempts to change the parameter p in an attempt to skew the results of the poll. Note that we cannot prevent her from decreasing p; she may always set b equal to the result of flipping a weighted coin that lands on heads with probability $1 > q \geq 1/2$, which will give an output distibution of $(pq + (1-p)(1-q), (1-p)q + (1-q)p)$ instead of $(p, 1-p)$. This has the effect of decreasing the weight of the responder's answer. Instead, we call a protocol responder immune if we prevent her from *increasing* p towards 1 (and so increasing her answer's weight). Consider the steps taken by the responder.

First, she takes a pile of chips and marks one. If she marks some other number of chips instead, this will throw off the final count; n will not correspond to the number of responders. In this case, the pollster can discard the results so far and start again. Note that if a large amount of cheating takes place, the pollster can potentially open the boxes after every responder. Then all honest responses are kept and dishonest ones discarded.

Next, she drops chips into the board one at a time. If she attempts to insert a paper clip or other object into the machine to force her chips along a certain path, this will be easily detected by the pollster, who watches the responder precisely to detect this type of action. If she puts all of her chips in one input, puts the chips in backwards, or otherwise exposes the value b, she does not disrupt the final tally. Having inserted a chip, she may watch it move through the board, but she may not affect its course. If, for example, she attempts to tilt the machine to one side to force a particular result, this will be apparent to the pollster, who will then discard the result. Other tampering, such as inserting a chip in the middle of the board, is prevented by the transparent cover.

3.2 Pollster Immunity

For proving that our scheme is pollster immune, the dishonest pollster's goal is to learn a responder's true answer with probability higher than p. We allow the pollster to deviate from the protocol or lie to the responder. We do not allow the pollster to interfere with the Plinko board while the responder is passing chips through it, though they may watch the proceedings.

First of all, the operation of the Plinko machine is simple enough that lying about either its security properties or the appropriate procedure should be unconvincing. For example, if the pollster claimed that only the true chip should be inserted into the machine, it should be obvious that this allows the pollster to observe the bit b directly. Any such deception should, with high probability, result in the responder refusing to use the board entirely.

Next we assume that the pollster faithfully gives instructions, yet still wishes to cheat. Suppose that he modifies the Plinko board such that the pegs do not cause left bounces and right bounces equally often, or so that one bounce affects the next. In this case, the deception may be detected by the test chips; the responder should only be convinced

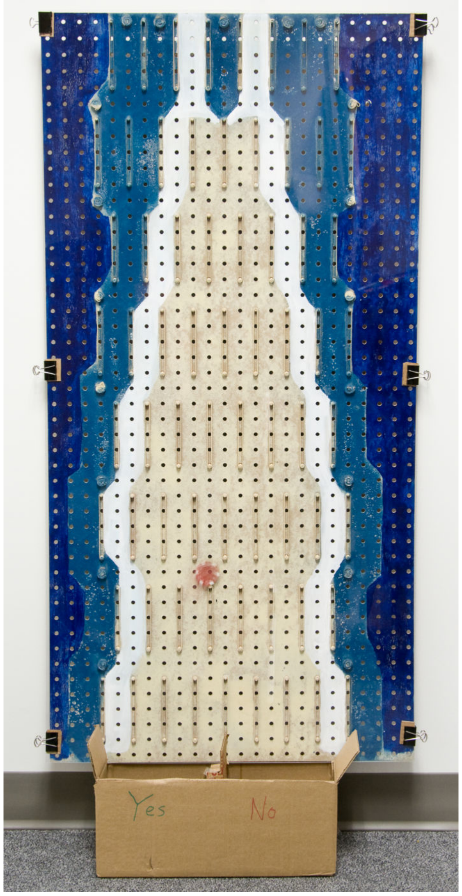


Photo credit: Chris Hughes

Figure 3: Our demonstration Plinko board. Its parameters are $h = 7$, $w = 8$, $d = 2$, and $p = 0.773$. Note the red chip falling through a vertical channel before hitting the peg below.

of the correctness of the machine if the bounces correspond to her intuition of how they should behave. If this testing is necessary, the responder can save the true chip and a single test chip until the end, observing all of the other test chips before committing to a value of b. If the responder puts these last two chips in opposite inputs, the pollster still learns nothing about b.

4 Estimating π

In terms of estimating π, our Plinko board works identically to Warner's original spinners. Given groups A and $\bar{A}$, then with probability p the responder answers the question "Do you belong to group A?", and the chip travels from an input slot to the corresponding most likely output slot. With probability $(1 - p)$, the responder actually answers the question "Do you belong to group $\bar{A}$?", as flipping an answer to the original question is equivalent to flipping the question and leaving the answer intact. So we may estimate π using the same formulae as Warner [10]. If there are n responders, and n_1 chips in the "Yes" output, then:

$$\begin{aligned}\hat{\pi} &= \frac{p-1}{2p-1} + \frac{n_1}{(2p-1)n} \\ Var_{\hat{\pi}} &= \frac{1}{n}\left[\frac{1}{16(p-1/2)^2} - (\pi - 1/2)^2\right] \\ &= V_r + V_s\end{aligned}$$

Where $V_r = \frac{(p-1/2)^{-2}-4}{16n}$ is the variance due to the randomness in the Plinko board, and $V_s = \frac{\pi(1-\pi)}{n}$ is the usual variance due to sampling.

All RRT-based schemes estimate π based on the assumption that all answers are truthful. Here, using a Plinko board instead of a spinner works to make that assumption more reasonable, but it does not alter the analysis. As in [10], we observe that larger values of p lead to lower values of V_r, and thus tighter confidence intervals; however, larger values of p also offer less privacy to participants, and thus suggest a lower likelihood of participants being truthful in their responses. For example, with our sample board ($d = 2$, $h = 7$) having $p = 0.773$, a result of $n_1 = 375$ "Yes" outputs out of $n = 1000$ samples yields a 95% confidence interval for π of 27% $\pm 5.5\%$. The same results on a board with $d = 6$ and $h = 7$ (and so $p = 0.992$) would give a 95% confidence interval of 37% $\pm 3.0\%$.

5 Limitations and Future Work

Although our method works well when the pollster and responder are in the same physical location, it does not have a clear parallel for polls conducted over the phone or by mail. In this case, it is more difficult to find reliable sources of randomness beyond the control of either party, and we leave this as future work.

We also note that the pollster may still be able to mount the following attack: by inserting an RFID tag into each chip, he may record when that chip enters the board, and through which input. Later, he may scan the chips in the output bins one by one to discover the identities of the marked chips, then compare this list to his log of entrance times. This would allow a pollster who knew the identities of the chips given to a particular responder to learn that responder's true answer. This is undesirable, though we note that all other random devices we have seen either allow the responder to abort early, or allow an attack with imperceptible transmitters and receivers leaking the random values as they are generated.

6 Conclusions

We have presented a secure scheme for in-person polling by using a physical noisy channel to permit the observed polled responses to vary from their original values. We showed that it is pollster and responder immune. We used a Plinko board as a noisy channel: it is sufficiently simple and familiar so that its security properties are highly intuitive, and lies about its properties are easy to detect. We gave methods for constructing a Plinko board for a desired probability of switching the response, using it to conduct a poll, and using the results to estimate the percentage of the population that gave each answer. We have shown how to estimate the variance of our answer, which allows the pollster to compute confidence intervals and estimate the sample size required. We constructed a demonstration board and found the construction to be simple and inexpensive. Though some issues remain unresolved, we have mitigated several main concerns by having simplified explanations, not requiring a third party, and widening the range of p values without sacrificing ease of use.

Acknowledgements

We thank Kevin Henry for assisting with the design and construction of our Plinko board, and Chris Hughes for photographing it. We gratefully acknowledge the support of NSERC and MITACS for funding this research.

References

[1] Ben Adida and Ronald L. Rivest. Scratch & Vote. In *Proceedings of the 2006 Workshop on Privacy in the Electronic Society*, pages 29–40. ACM Press, October 2006.

[2] Andris Ambainis, Markus Jakobsson, and Helger Lipmaa. Cryptographic randomized response techniques. *CoRR*, cs.CC/0302025, 2003.

[3] Kevin Fisher, Richard Carback, and Alan T. Sherman. Punchscan: Introduction and System Definition of a High-Integrity Election System. In *Proceedings of the 2006 Workshop on Trustworthy Elections*.

[4] N. S. Mangat. An improved randomized response strategy. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):93–95, 1994.

[5] N. S. Mangat and R. Singh. An alternative randomized response procedure. *Biometrika*, 77:439–442, 1990.

[6] Tal Moran and Moni Naor. Polling with Physical Envelopes: A Rigorous Analysis of a Human-Centric Protocol. *Advances in Cryptology - EUROCRYPT 2006*, pages 88–108, 2006.

[7] PromoQuip. Plinko boards specs and online ordering. http://www.promoquip.com/plinko.htm. Accessed March 2009.

[8] Peter Y. A. Ryan and Thea Peacock. Prêt à Voter: a systems perspective. Technical Report CS-TR-929, University of Newcastle upon Tyne, 2005.

[9] N. J. Scheers and C. Mitchell Dayton. Improved Estimation of Academic Cheating Behaviour Using the Randomized Response Technique. *Research in Higher Education*, 26(1):61–69, March 1987.

[10] Stanley L. Warner. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *Journal of the American Statistical Association*, 60(309):63–69, March 1965.