# The Number of Partitions in Pollard Rho

Simon R. Blackburn and Sean Murphy

Technical Report
RHUL–MA–2011–11
31 March 2011


Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England

http://www.rhul.ac.uk/mathematics/techreports

# The Number of Partitions in Pollard Rho

Simon R. Blackburn* and Sean Murphy
Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, United Kingdom

March 31, 2011

**Abstract**

This technical report was originally a note dated 1 May 1998, not intended for publication. This version is identical to the original, though we've taken the opportunity to update references and correct a minor typo.

## 1 Introduction

Let $G$ be a cyclic group of prime order $n$. Let $g$ generate $G$ and let $h \in G$. Pollard's rho method [1] is a $O(\sqrt{n})$ algorithm which finds an integer $x$ such that $g^x = h$ (the discrete log of $h$ to the base $g$).

The classical variant of Pollard's algorithm relies on choosing a partition of $G$ into 3 parts, and then defining a function $f : G \to G$ by selecting a different formula on each part of the partition in such a way that the relationship between the log of $f(a)$ and the log of $a$ is known (we omit the details as they are standard). The algorithm operates by choosing a point $s$ uniformly at random from $G$ and then computing elements of the set $\rho = \{f^i(s) : i \geq 0\}$ in turn. The expected running time of the algorithm is proportional to the expected order of the set $\rho$.

*This author was supported by an E.P.S.R.C. Advanced Fellowship

The function $f$ is usually modelled as a random function from $G$ to itself. In this case, the expected order of the set $\rho$ may be modelled as the expected time of first collision in a random walk on a set of size $n$. In other words, suppose we have a random variable $X$ with the property that $P(X = \ell \mid X \geq \ell) = \ell/n$ for all $\ell \in \{1, 2, \ldots, n\}$. Then $E(X)$ is the expected order of $\rho$ under this model. There are arguments that indicate that $E(X)$ is approximately $c\sqrt{n}$, where $c = \sqrt{\pi/2}$.

In practice (see the paper of Teske [2]) the algorithm seems to take longer than this analysis would indicate. Taking $k$ parts in the partition, where $k$ is larger than 3 ($k = 16$ or $k = 20$), improves the experimental expected running time of the algorithm. Why does increasing the number of parts of the partition lead to an improvement? This note gives an explanation for this phenomenon. Our argument is non-rigorous, but we feel that it can be made so.

## 2 A New Model for the Function $f$

Let $f$ be a function chosen by partitioning $G$ into $k$ parts of approximately equal size, and by choosing a different rule on each part of the partition. For all $i \in \{1, 2, \ldots, k\}$, let $f_i$ be defined to be the restriction of $f$ to the $i$th partition. We assume that $f_i$ is always an injection. As an example, we can take $f(a) = aa_i$ for all $a$ in the $i$th part of the partition, where $a_1, a_2, \ldots, a_k$ are fixed elements of $G$ (whose logarithms are known).

Define a directed graph whose vertices correspond to elements of $G$ and where there is a directed edge from $a$ to $f(a)$ for all $a \in G$. A vertex $a \in \Gamma$ has in-degree at most $k$ (as there is an incoming edge associated with part $i$ of the partition if and only if $a \in \text{im } f_i$). Since $|\text{im } f_i|$ is approximately $n/k$, we have that an incoming edge associated with part $i$ occurs with probability $1/k$ if $a$ is chosen uniformly at random from $G$. Hence we would expect (and assume) the indegree of a random vertex to behave like the appropriate binomial distribution — i.e. the probability that a vertex chosen at random has indegree $d$ should be very close to

$$p_d = \binom{k}{d} \left(\frac{1}{k}\right)^d \left(\frac{k-1}{k}\right)^{k-d}.$$

Of course, the expected in-degree of a vertex is precisely 1, since every

vertex has out-degree precisely 1. What is the expected in-degree of a vertex in the set $\rho$? Apart from the first element $s$ of $\rho$, we expect (and assume) that the expected in-degree of a vertex in $\rho$ may be modeled by the expected in-degree of the vertex $f(s)$, where $s$ is chosen uniformly at random. Vertices with indegree $d$ are $d$ times as likely to occur as the value of $f(s)$ than vertices with in-degree 1, and so we expect that the probability of a vertex in $\rho$ having degree $d$ should be approximated by

$$dp_d / \left( \sum_{\ell=0}^{k} \ell p_\ell \right)$$

Hence the expected in-degree of a vertex in $\rho$ should be

$$\begin{aligned} \sum_{d=0}^{k} d \left( dp_d / \left( \sum_{\ell=0}^{k} \ell p_\ell \right) \right) &= \sum_{d=1}^{k} d^2 \binom{k}{d} (k-1)^{k-d} / \sum_{d=1}^{k} d \binom{k}{d} (k-1)^{k-d} \\ &= 1 + \frac{k-1}{k}. \end{aligned}$$

Let $\rho_i$ be the first $i$ vertices in $\rho$. Now, the probability that the length of $\rho$ is equal to $i$, given that $\rho$ has length at least $i$ may be estimated as equal to the proportion of all edges from vertices not in $\rho_i$ that lead to a vertex in $\rho_i$. The number of edges from vertices not in $\rho_i$ is equal to $n-i$. The number of these edges that lead to a vertex in $\rho_i$ may be estimated to be $i(d-1)$, where $d$ is the expected in-degree of a vertex in $\rho$. Thus the probability that the length of $\rho$ is equal to $i$, given that $\rho$ has length at least $i$ may be estimated to be $i(d-1)/(n-i)$. Now, $\rho$ never gets much larger than $O(\sqrt{n})$, so we may approximate this probability by $i(d-1)/n$. We estimated $d$ above as $d = 1 + \frac{k-1}{k}$, so we have that this probability is approximately $i/(\frac{k}{k-1}n)$. But this is just the probability that $\rho$ has length $i$ given that is has length at least $i$ in the situation of the original random function model for $f$ when $|G| = \frac{k}{k-1}n$. So we would expect that the average length of $\rho$ in our new model to be about $c\sqrt{\frac{k}{k-1}n}$, where $c = \sqrt{\pi/2}$ as before.

In summary, for a function $f$ derived from $k$ partitions, we would expect that the algorithm would take a factor of $\sqrt{\frac{k}{k-1}}$ longer than the random function model would predict. When $k = 3$, this factor is 1.225, which agrees with the experimental data presented in Teske [2].

# References

[1] J.M. Pollard, 'Monte Carlo methods of index computation (mod $p$)', *Math. Comp.* 32 (1978), 918-924.

[2] E. Teske, 'Speeding up Pollard's rho method for computing discrete logarithms', in *Algorithmic Number Theory* (J.P. Buhler, ed), Lecture Notes in Computer Science 1423 (Springer, Berlin, 1998), 541-554.