Collision of Random Walks and a Refined Analysis of Attacks on the Discrete Logarithm Problem

Shuji Kijima[1] and Ravi Montenegro[2](✉)

[1] Graduate School of Information Science and Electrical Engineering,
Kyushu University, Fukuoka 819-0395, Japan
kijima@inf.kyushu-u.ac.jp

[2] Department of Mathematical Sciences, University of Massachusetts Lowell,
Lowell, MA 01854, USA
ravi_montenegro@uml.edu

Abstract. Some of the most efficient algorithms for finding the discrete logarithm involve pseudo-random implementations of Markov chains, with one or more "walks" proceeding until a collision occurs, i.e. some state is visited a second time. In this paper we develop a method for determining the expected time until the first collision. We use our technique to examine three methods for solving discrete-logarithm problems: Pollard's Kangaroo, Pollard's Rho, and a few versions of Gaudry-Schost. For the Kangaroo method we prove new and fairly precise matching upper and lower bounds. For the Rho method we prove the first rigorous non-trivial lower bound, and under a mild assumption show matching upper and lower bounds. Our Gaudry-Schost results are heuristic, but improve on the prior limited understanding of this method. We also give results for parallel versions of these algorithms.

1 Introduction

Given a cyclic group $G = \langle g \rangle$ and an element $h \in G$, the discrete-logarithm problem asks to find a solution x to $h = g^x$. Shoup showed that for a generic cyclic group this requires $\Omega(\sqrt{|G|})$ group operations [17], although this bound can be beaten for many representations of such groups. The discrete-logarithm problem over a random group of elliptic curves seems to be as hard as this lower bound, which has led to its use in cryptosystems.

Several methods have been proposed which use a pseudo-random "walk" to achieve heuristic run time equal to Shoup's lower bound. Each step of these walks will involve a single group operation and so the number of steps (run time) will equal the number of group operations until discrete logarithm is found, aside from a small amount of pre-computation. In this paper we give a fairly general method for understanding the performance of such methods, and in particular

R. Montenegro—Supported by a Japan Society for Promotion of Science (JSPS) Fellowship while a guest at Kyushu University.

J. Katz (Ed.): PKC 2015, LNCS 9020, pp. 127–149, 2015.
DOI: 10.1007/978-3-662-46447-2_6

for understanding the extent to which they will be slower than predicted by simple heuristics. We use our method to show very precise estimates on run time of three such methods: Pollard's Rho, Pollard's Kangaroo, and Gaudry-Schost. These methods have been used for attacks on problems of cryptographic significance. For instance, a parallelized Pollard's Rho method was used in an attack on Certicom's challenge problem ECC2K-130 [2], while an attack based on Gaudry-Schost was used to break a proposed EMVco protocol to replace the chip-and-pin system used in over 1.6 billion payments cards [3].

1.1 Description of Algorithms

Before describing our results we review the algorithms being considered. Methods of detecting collisions, such as via distinguished points or Floyd's cycle finding method, will not be discussed.

For Pollard's Rho, partition G into 3 roughly equal sized pieces S_1, S_2, S_3, and define an iterating function

$$F(X) = \begin{cases} Xg & \text{if } X \in S_1, \\ Xh & \text{if } X \in S_2, \\ X^2 & \text{if } X \in S_3. \end{cases}$$

Let $X_0 = h = g^x$ and repeatedly iterate with $X_{i+1} = F(X_i)$. Continue until the first time that some $X_i = X_j$, known as a "collision." If we keep track of the exponent $X_i = g^{a_i + b_i x}$ then $g^{a_i+b_i\,x} = g^{a_j+b_j\,x}$. The discrete logarithm is then $x \equiv (a_i - a_j)(b_j - b_i)^{-1} \mod |G|$, except in the rare degenerate case when $b_i \equiv b_j \mod |G|$. Teske suggests an "additive" version that is faster in practice. For a fixed integer r define r step types s_1, s_2, …, s_r by choosing α_k, β_k uniformly at random from $\{0, 1, \ldots, |G| - 1\}$ and setting $s_k = g^{\alpha_k} h^{\beta_k} = g^{\alpha_k + \beta_k x}$. Then take r partitions S_1, S_2, …, S_r, set $F(X) = X\,s_k$ on S_k, and proceed as before. One way to parallelize is to start M processors at different randomly chosen states: $g^\alpha h^\beta = g^{\alpha+\beta x}$ with $\alpha, \beta \in [0, |G| - 1]$. These may be re-randomized every time a distinguished point is encountered (see Gaudry-Schost below and [19]).

Pollard's Kangaroo method applies when it is known that $x \in [a, a + N)$ for some $a, N \leq |G|$. Set $X_0 = h = g^x$ and $Y_0 = g^{a+\lfloor N/2 \rfloor}$. Take $d + 1$ partitions S_0, S_1, …, S_d, set $F(X) = X\,g^{2^k}$ on S_k, and repeatedly iterate both processes with $X_{i+1} = X_i\,F(X_i)$ and $Y_{i+1} = Y_i\,F(Y_i)$. Once some $X_i = Y_j$, say at $X_i = g^{x+\alpha}$ and $Y_j = g^{a+\lfloor N/2 \rfloor + \beta}$, then $x \equiv a + \lfloor N/2 \rfloor + \beta - \alpha \mod |G|$, and the discrete logarithm is found. The processes X_i and Y_j are known as the wild and tame kangaroos respectively.

In the Gaudry-Schost method the discrete logarithm x is known to lie in a hypercube $[a, b]^n$ with volume $N = (b - a + 1)^n$. We discuss only the $n = 1$ version here, although our technique applies in higher dimensions as well. Let A be a region centered at the unknown discrete logarithm x, let B be a duplicate of this region but centered at a predetermined value within the hypercube such as the centerpoint, and let D be a set of *distinguished points* covering roughly a θ

fraction of group elements. For a pre-specified integer r and average step size m choose r step types $s_1, s_2, \ldots, s_r$ uniformly at random from $[1, 2m)$, and partition the space into r pieces $S_1, S_2, \ldots, S_r$. Use iterating function $F(X) = X\, g^{s_i}$ if $X \in S_i$, except for the θ fraction of the time that X is a distinguished point and F transitions to a point chosen uniformly at random in A or B. Proceed until some $X_i = Y_j$, say with $X_i = g^{x+\alpha}$ and $Y_j = g^{\beta}$, at which point the discrete logarithm is $x \equiv \beta - \alpha \mod |G|$. This will usually be in $A \cap B$, but could be slightly outside this region. One way to parallelize is to start $M/2$ processors at different randomly chosen states in A, and $M/2$ in B.

1.2 Heuristic Run Time

The attacks just described involve pseudo-random processes on G which proceed until a *collision*: either a single walk proceeds until it visits a state (group element) it has previously been to, or two walks proceed until each has visited a common state (group element). If we treat these as truly random processes then there are natural heuristic arguments for their run time.

Pollard's Rho resembles a random walk which proceeds until some state is visited twice. If each transition were a uniform random sample from G then the birthday paradox would suggest run time of $\sqrt{\frac{\pi}{2}|G|} \approx 1.25\sqrt{|G|}$. However, because the process only has 3 transition types, consecutive states are highly dependent and the true run time is about 30% slower. Teske's additive version is significantly slower when there are $r = 3$ transition types, but is nearly as good as the birthday heuristic when r is large, e.g. $r = 16$. Several improvements have been made on this basic heuristic, see Section 2.1. The parallel version with M processors generates samples M times faster, and so is M times faster.

Pollard's Kangaroo resembles two random walks which proceed until they visit some common state. Let $d \sim \log_2 \sqrt{N} + \log_2(\log_2 \sqrt{N}) - 2$ be such that the average transition size is $m = \frac{1}{d+1}\sum_{k=0}^{d} 2^k \approx \frac{1}{2}\sqrt{N}$. After a warmup of around $\mathbb{E}T = \mathbb{E}\frac{|X_0 - Y_0|}{m} = \frac{N}{4m} = \frac{1}{2}\sqrt{N}$ steps, the walk starting with smaller exponent of $X_0 = g^x$ and $Y_0 = g^{a+N/2}$ will have caught up to the initial location of the other walk. Each walk visits a $1/m$ fraction of states, so at each subsequent step there is probability $p \approx 1/m$ of a "collision," for expected runtime of about $2(\mathbb{E}T + m) \approx 2\sqrt{N}$. However, because the process only has $(d + 1)$ transition types there are dependencies and the probability of a collision varies significantly from step to step, with probability often 0.

The Gaudry-Schost method resembles two random walks, one in A and one in B, where each step produces a sample from $A \cap B$ with probability roughly $\frac{|A \cap B|}{|A|}$. A generalized birthday problem [14] suggests that each walk should take $\frac{1}{2}\sqrt{\pi\,|A \cap B|}$ samples from $A \cap B$ until collision, for an expected run time of $2\,\frac{|A|}{|A \cap B|}\,\frac{1}{2}\sqrt{\pi\,|A \cap B|}$ transitions. But again, consecutive states are highly dependent unless a distinguished point was just visited. Once again, the parallel version generates samples M times faster.

The largest flaw in each heuristic is that consecutive states are highly dependent. One solution might be if Rho, for instance, used an iterating function

$F : G \to G$ that outputs a pseudo-random uniform sample from the entirety of G. However, computing random values of $g^\alpha h^\beta$ is slow. Even if in an average of only two group operations sufficed, finding the discrete logarithm would take $2 \times 1.25\sqrt{|G|}$ group operations, versus $1.6\sqrt{|G|}$ for Pollard's Rho and under $1.3\sqrt{|G|}$ for Teske's additive version. So the dependencies in these algorithms are an important component of their fast run time. An improved method for understanding the effects of dependencies could thus help to minimize their negative effects.

1.3 New Results

Each algorithm considered here is entirely deterministic once the partition function (hash) has been chosen. Indeed, the deterministic nature is necessary for efficient detection of collisions. However, the hash is usually chosen to "look" random, and so all attempts to explain these algorithms have treated the iterative process as if it were entirely random until the first collision. Our results will make this assumption as well, even those we describe as "rigorous." Since our concern is with when that collision occurs, not what happens after it, we also treat these as fully random walks, even after collision. As such we will use the language of random walks, so that "state" refers to a group element and "run time" refers to the number of iterations taken. After some precomputation each iteration requires exactly one group operation, and so run time will be equivalent to counting group operations, except when a distinguished point is hit in Gaudry-Schost.

The heuristic arguments just described neglect dependencies between consecutive states. One way of avoiding this problem is to only consider a subset of states all of which are τ steps apart, where τ denotes the number of steps required to lose this dependency, so that X_j is nearly independent of X_i when $|j - i| \geq \tau$. However, this would give results which are very weak since a large fraction of possible collisions are being ignored.

A different approach is to try to measure the extent of the dependency. Consider how many times two independent random walks can be expected to collide (visit common states) if they start at the same state and proceed τ steps until they have lost their initial dependency. We find that this quantity alone is sufficient to explain the extent to which Pollard's Kangaroo and Pollard's Rho fail to match heuristic bounds. It also explains the vast majority of the slow down for Gaudry-Schost, although boundary effects along $\partial(A \cap B)$ also come into play. The precise quantity we consider is the following:

Definition 1. *The* collision number $\mathcal{C}_\tau$ *is the expected number of collisions when two independent copies of a random walk start at the same state, chosen uniformly at random, and proceed for τ steps.*

Consider the Kangaroo method. This is generally thought to be well understood, in the sense that the heuristic of $2\sqrt{N}$ matches asymptotic behavior. However, even on groups as large as $|G| = 10^{12}$ the Kangaroo method runs about 3% slower than predicted by heuristic. We use the collision number to prove a bound which more-or-less eliminates this error (see Figure 1).

Theorem 2 (Pollard's Kangaroo). *Given a cyclic group G, the Kangaroo method with power-of-two jumps for computing the discrete logarithm in an interval of size $N = o(|G|)$ has expected run time*

$$2\,\mathbb{E}\min\{k : \exists i,j \le k,\, X_i = Y_j\} = \left(2 + \frac{2}{\log_2 N} + \frac{14}{(\log_2 N)^2} + O((\log N)^{-3})\right)\sqrt{N}\,.$$

This extends to transitions other than powers of two if $2 + \frac{2}{\log_2 N} + \cdots$ is replaced by $1 + \mathcal{C}_\tau$.

For Pollard's Rho method we show the first rigorous non-trivial lower bound on runtime.

Theorem 3 (Pollard's Rho). *Given a cyclic group G of prime order N, the Rho method with r step types has expected run time of*

$$\mathbb{E}\min\{i : \exists j \le i-\tau),\, X_i = X_j\} \ge (1 + O(1/N))\sqrt{\frac{\pi}{2}\,\frac{N}{1-1/r}}$$

where $\tau = O(\log^2 N)$ for Pollard's process and $\tau = O(N^{2/(r-1)})$ for Teske's.

This neglects the $o(1)$ fraction of potential collisions $X_i = X_j$ with $i - j < \tau$, except for Teske's process with $r \le 5$.

Under a fairly mild assumption we can determine the precise run time.

Heuristic 4 (Pollard's Rho) *Given a cyclic group G of prime order N, Pollard's Rho method has expected run time of*

$$\mathbb{E}\min\{i : \exists j < i,\, X_i = X_j\} = (1.62555 + O(1/N))\sqrt{N}\,.$$

Teske's additive version with $r \ge 6$ step types has

$$\mathbb{E}\min\{i : \exists j \le i,\, X_i = X_j\} = (1 + O(1/N))\sqrt{\frac{\pi}{2}\,\frac{N}{1-(1/r+1/r^2+2/r^3+O(1/r^4))}}\,.$$

Parallel versions with M threads will take $1/M$ times as long.

Simulation data matches this to 4 decimals points , so the heuristic is almost certainly correct.

The final process we consider is the Gaudry-Schost method. This has boundary effects which complicate any attempt at a rigorous proof, and there are too many variants to analyze them all here. In order to present the ideas without complicating things excessively we have chosen to analyze one of the simpler cases [6].

Heuristic 5 (Gaudry-Schost) *Galbraith and Ruprai's Gaudry-Schost method for computing the discrete logarithm on an interval of size N with r step types, distinguished point probability θ at each state, and average step size $m = c\,\theta\,N$ with c small (e.g. $c = 0.01$), has expected run time*

$$\begin{aligned} &2\mathbb{E}\min\{k : \exists i, j \leq k,\, X_i = Y_j\} \\ &= (1 + O(1/N))\,\frac{2}{3^{1/2}}\sqrt{\pi\frac{N}{1 - (1/r + 1/r^2 + 2/r^3 + O(1/r^4))}}\,. \end{aligned}$$

Parallel versions with M threads will take $1/M$ times as long.

This indicates a slowdown by $\sqrt{\mathcal{C}_\tau} = 1/\sqrt{1 - (1/r + 1/r^2 + 2/r^3 + O(1/r^4))}$ over previous heuristics based on the birthday problem. We test our method on a more complicated case of Gaudry-Schost, namely Galbraith, Pollard, and Ruprai's [5] improved 3 walk Gaudry-Schost method for discrete logarithm on an interval of width N. Our heuristic predicts the runtime within 0.3% of that found in simulations.

The paper proceeds as follows. In Section 2 we discuss past results on collision times and also give an overview of our new method for studying collision times. We apply this method to Pollard's Kangaroo in Section 3, to Pollard's Rho in Section 4, and to Gaudry-Schost in Section 5. Section 6 includes discussion on computation of $\mathcal{C}_\tau$, while Section 7 consolidates our simulation data confirming the high degree of accuracy in our results.

2 Methods of Studying Collision Time

The attacks considered in this paper depend on iterative processes which proceed until some group element has been visited twice, a "collision." In Section 1.2 we gave simple heuristics for understanding the time until the first collision. In Section 1.3 we justified treating the attacks as if they involved random walks. Under this assumption several improvements have been made on the basic heuristic arguments, and we discuss here both those improvements and our new approach to studying collision time.

We use some notation here that may be unfamiliar: $f = o(g)$ if $\lim_{x\to\infty}\frac{f(x)}{g(x)} = 0$, while $f = O(g)$ if $\lim_{x\to\infty}\frac{f(x)}{g(x)} \leq C$ for some constant C, and $f = O^*(g)$ indicates that logarithmic terms are being ignored, e.g. if $f = O(x^3 \log x)$ then $f = O^*(x^3)$.

2.1 Past Work

Of the three methods we consider, Pollard's Rho has been studied most heavily. A heuristic based on the birthday problem suggests that it will take an average of $\sqrt{\frac{\pi}{2}\,N} = 1.2533\sqrt{|G|}$ steps until a collision. However, experimental data finds the run time to be slower than this, and sometimes significantly so. Blackburn and Murphy [4] borrow an idea of Brent and Pollard to give an improved heuristic, that for a process with r step types the collision time will be $\sqrt{\frac{\pi}{2}\,\frac{N}{1-1/r}}$. Teske gives a heuristic suggesting that her additive version has run time $O(\sqrt{N})$ when there are $r \geq 5$ step types [18]. The first rigorous result for a Rho method is Miller and Venkatesan's proof of order $O^*(\sqrt{N})$ for Pollard's Rho [12]. Kim, Montenegro, Peres and Tetali [10] improve this to $O(\sqrt{N\,\mathcal{C}_\tau})$, with Pollard's Rho having expected collision time of $\leq (52.5+o(1))\sqrt{N}$. Bailey et al. [1] extend Blackburn and Murphy's method to the case where steps do not all have equal probabilities. Bernstein and Lange [2] take a very different approach to the problem and yet also arrive at our Heuristic 4 for the special case of Teske's additive walks.

Pollard's Kangaroo method on an interval $[a, a+N)$ is based on a principle known as the Kruskal Count, which suggests a run time of $(2+o(1))\sqrt{N}$. Pollard gives a very convincing argument that $(2+o(1))\sqrt{N}$ steps suffice, although it was not quite rigorous. Montenegro and Tetali give the first rigorous result that $(2+o(1))\sqrt{N}$ steps suffices [13]. However, their upper and lower bounds do not agree on the error term, or even whether the run time is greater or less than $2\sqrt{N}$.

For the Gaudry-Schost method nothing has been shown rigorously, and simulations disagree somewhat with heuristic. This method is complicated by having several variables: the number of generators, the average step size, and the number of distinguished points. Our result is thus the first attempt at a bound better than what can be obtained by a simple generalized birthday problem, and yet it is quite accurate, predicting runtime within 0.3% of what we find in simulations.

2.2 Our Approach

We take a similar approach to each algorithm. First, choose the appropriate heuristic from Section 1.2. This would be more-or-less rigorous if the probability of a collision between every pair of states X_i and X_j were independent, but this is clearly not the case.

We revise the heuristic by replacing each individual state X_i by a long segment S_i of some L consecutive states, preceded by a short randomization segment R_i of τ states to ensure that S_i is independent of the earlier segments. In particular, let $a_0 = 0$, $b_0 = a_0 + \tau$, $a_\ell = b_{\ell-1} + L$, $b_\ell = a_\ell + \tau$.

$$\begin{aligned} R_\ell &= \{X_{a_\ell}, \ldots, X_{b_\ell}\} \\ S_\ell &= \{X_{b_\ell}, \ldots, X_{a_{\ell+1}}\} \end{aligned} \tag{1}$$

By construction segments S_i and S_j will be independent when $i \neq j$. If $\tau \ll L$ almost no collisions will involve randomization segments R_i, while if L is much less than the expected collision time almost no collisions will involve a segment S_i with itself, leaving almost all collisions to be between distinct segments S_i and S_j. It is generally not hard to modify the heuristic of Section 1.2 to determine a new and precise estimate of collision time. The hard part is in adding rigor.

First, although $\mathcal{C}_\tau$ is fairly easy to estimate numerically it is very difficult to find τ and $\mathcal{C}_\tau$ rigorously. Second, it can be difficult to show that there is indeed a negligible chance of collisions between a segment S_i and itself, or of a collision involving an R_i segment. Most of the technical work required for rigor is left to the full version of this paper or cited from prior research, as it is tedious and not very enlightening.

The simplest approach to parallelization involves starting M threads at M independent initial points, and recording visits to distinguished points until some distinguished point is visited a second time. For fixed values of L and τ each thread will generate a new nearly-independent segment every $(\tau + L)$ steps, and so this form of parallelization produces segments M times faster, and expected collision time is $1/M$ times as long.

Our approach is inspired by our past work [10,13], and indeed we borrow several of the more tedious results from those papers. However, a critical difference is that we now partition the walk by design, with a conscious goal to rework prior heuristics. In our past work the walk was partitioned as well, but that was an artifact of the proof failing to work up to the full collision time. As a result the partitioning felt unnecessary, the method of proof was harder to follow, and the results were less precise than obtained in this paper.

3 Pollard's Kangaroo Method

We begin by studying Pollard's Kangaroo method, as the argument is somewhat simpler and sharper than in the Rho or Gaudry-Schost cases.

Collision in the random walk is equivalent to collision in the exponent of g, i.e. $g^\alpha = g^\beta$ iff $\alpha \equiv \beta \mod |G|$. This induces an additive form of the process: let $X_0 = x$, $Y_0 = a + N/2$, and transitions are of the form $X_{i+1} = X_i + 2^k \mod |G|$ and $Y_{i+1} = Y_i + 2^\ell \mod |G|$, where $k, \ell \in \{0, 1, \ldots, d\}$. Furthermore, we are most interested in the case when N is much smaller than the group size $|G|$, and so wrap-around effects $\mod |G|$ can safely be neglected. This lets us simplify further: assume that $|X_0 - Y_0| \leq N/2$ and take iterations of the form $X_{i+1} = X_i + 2^k$ and $Y_{i+1} = Y_i + 2^\ell$. The Kangaroo process is then a monotone increasing walk on the integers $\mathbb{Z}$ with $X_{i+1} = X_i + 2^k$, $Y_{i+1} = Y_i + 2^\ell$. Due to this simplification, our upper bound will be valid for all N, but the lower bound holds only when $N = o(|G|)$.

The goal is to determine the expected time of the first collision between these processes:

$$\mathbb{E} \min\{k : \exists i, j \leq k,\, X_i = Y_j\}\,.$$

Because both processes are run simultaneously, the total number of steps taken is twice this.

The non-rigorous part of the heuristic in Section 1.2 is the claim that collisions occur with probability $p \approx 1/m$, and so $p^{-1} \approx m$ steps are required until a collision. We replace this by a claim that there is a $p = (1 + o(1))\, L/m^2$ probability of segment S_i including a state visited by the tame kangaroo, and so an average of p^{-1} segments are needed until a collision. The expected run time of the Kangaroo method is then $\frac{N}{4m} + (\tau + L)p^{-1}$.

This argument neglects potential collisions involving the R_i segments, and so it is only an upper bound. To show a lower bound we set $L = |S_i| = \sqrt[4]{N}$ and $\tau = |R_i| = O((\log N)^6)$, so that $\tau = o(L)$ and the R_i segments are involved in only a $o(1)$ fraction of potential collisions. We show in the full version of the paper that the R_i segments have a negligible probability of being the location of the first collision.

Why this value for τ? The re-randomization portion R_i is intended to make the probability of a collision in S_i be independent of the outcome of earlier segments. One approach to this would be to have τ be the mixing time, i.e. the number of steps required to produce a uniform random sample from G. However, this is too pessimistic as the Kangaroo method for $N \lll |G|$ might even solve for the discrete logarithm in fewer steps than the mixing time. Instead we require a local mixing property. The follow property, defined by Montenegro and Tetali [13], suffices:

Definition 6. *Consider two independent instances of the same monotone increasing Markov chain on the infinite state space $\mathbb{Z}$, i.e. walks $X_0, X_1, \ldots$ and $Y_0, Y_1, \ldots$ such that $\forall i : X_{i+1} > X_i$ and $\forall j : Y_{j+1} > Y_j$. If the Markov chain has average step size m then the* intersection mixing time *$T(\epsilon)$ is the smallest integer with*

$$\forall i \geq T(\epsilon),\, \forall Y_0 \leq X_0 :\ \frac{1-\epsilon}{m} \leq \mathsf{P}\left(X_i \in \{Y_0, Y_1, \ldots, Y_k, \ldots\}\right) \leq \frac{1+\epsilon}{m}\,.$$

In our analysis the walks in Definition 6 will correspond to the wild and tame kangaroos defined at the beginning of this section. Intersection mixing time was studied in [13] where Lemma 3.1 shows that the Kangaroo walk with steps $\{2^k\}_{k=0}^{d}$, when treated as a monotone walk on $\mathbb{Z}$, satisfies $T(2/(d+1)) \leq 64(d+1)^5$. Mixing type results typically have a dropoff similar to $T(\epsilon) = T(1/e)\log(1/\epsilon)$. Since $d \sim 0.5\,\log_2 N + O(\log\log N)$ this suggests that

$$T(1/N) = O((d+1)^5 \frac{\log N}{\log D}) = O((d+1)^6) = O((\log N)^6)\,.$$

Indeed, the proof in [13] can be easily modified to show this.

It remains to determine the probability that a collision occurs in a segment S_k. The following relation will be used to show this. Given a non-negative random variable Q:

$$\mathsf{P}\left(Q > 0\right) = \frac{\mathbb{E}Q}{\mathbb{E}(Q \mid Q > 0)}\,.$$

Let CS_i denote the number of collisions between the tame walk and segment S_i in (1). Each state within S_i has probability $(1 + O(1/N))/m$ of colliding with the tame walk. It follows from additivity of expectation that

$$\mathbb{E}\mathrm{CS}_i = (1 + O(1/N))\,\frac{L}{m}\,.$$

We next bound the conditional $\mathbb{E}(\mathrm{CS}_i \mid \mathrm{CS}_i > 0)$. The first collision is part of a sequence of $\mathcal{C}_\tau$ collisions on average, after which each step has probability at most $(1+1/N)/m$ of colliding with the tame walk.

$$\mathbb{E}(\mathrm{CS}_i \mid \mathrm{CS}_i > 0) \leq (1+1/N)\left(\mathcal{C}_\tau + \frac{L-\tau}{m}\right)$$

To lower bound the expectation, observe that each step of the walk in S_i is equally likely to have a collision, and so the probability that $X_i \in S_i$ is the first collision is decreasing in i. As such, with probability at least $1-\tau/L$ the collision occurs before the final τ states, in which case it is part of a sequence of $\mathcal{C}_\tau$ collisions on average, potentially followed by even more. Then

$$\mathbb{E}(\mathrm{CS}_i \mid \mathrm{CS}_i > 0) \geq (1-1/N)\left(1-\frac{\tau}{L}\right)\mathcal{C}_\tau\,.$$

Combining these various equalities leads to the conclusions:

$$\Rightarrow\; \mathbb{E}(\mathrm{CS}_i \mid \mathrm{CS}_i > 0) = \left(1+O\left(\frac{L}{m}+\frac{\tau}{L}+\frac{1}{N}\right)\right)\mathcal{C}_\tau$$

$$\Rightarrow\; p = \mathsf{P}\left(\mathrm{CS}_i > 0\right) = \left(1+O\left(\frac{L}{m}+\frac{\tau}{L}+\frac{1}{N}\right)\right)\frac{L}{m\,\mathcal{C}_\tau} \tag{2}$$

The identity for $\mathsf{P}\left(\mathrm{CS}_i > 0\right)$ is independent of the outcome on earlier segments, up to a small error due to the big-O term, and so as was discussed earlier the number of segments until a collision is p^{-1}. Let $L = \sqrt[4]{N}$, and recall that $\tau = O(\log^6 N)$ and $m = \Theta(\sqrt{N})$. The expected collision time of the original process is

$$(L+\tau)p^{-1} = (L+\tau)\left(1+O\left(\frac{L}{m}+\frac{\tau}{L}+\frac{1}{N}\right)\right)\frac{m\,\mathcal{C}_\tau}{L} = (1+O^*(1/\sqrt[4]{N}))\,m\,\mathcal{C}_\tau\,.$$

We leave it to the full version of the paper to prove that the answer does not change when potential collisions in the re-randomization segments R_i are considered.

When $m = \frac{1}{2}\sqrt{N}$ the expected number of transitions by the wild kangaroo is

$$\mathbb{E}\frac{|X_0-Y_0|}{m} + (1+O^*(1/\sqrt[4]{N}))\,m\,\mathcal{C}_\tau = \frac{1+\mathcal{C}_\tau+O^*(1/\sqrt[4]{N})}{2}\sqrt{N}\,.$$

The tame kangaroo travels an equal number of steps. Counting both kangaroos gives Theorem 2.

In Figure 1 we compare our Theorem 2 to the old heuristic of $2\sqrt{N}$ and simulation data with an (absolute) margin of error of $\pm 0.01\sqrt{N}$. These show our bound to be very accurate when $N > 1000$.

4 Pollard's Rho Method

The analysis for the Rho method is not much different, but a bit more preliminary work is required, and the result will not be as precise. Our solution focuses on the case when each step type has equal probability, with a few comments at the end about generalizing to the non-uniform case. We focus on the non-parallel case because the parallel case follows immediately from this, as discussed in Section 2.2.

Pollard's Rho is equivalent to an additive walk on exponents which starts at some X_0 with $0 \leq X_0 < |G|$, and has transitions $\mathsf{P}(X_{i+1} = X_i + 1) = \mathsf{P}(X_{i+1} = X_i + x) = \mathsf{P}(X_{i+1} = 2X_i) = 1/3$. Teske's version has transitions $\mathsf{P}(X_{i+1} = X_i + s_k) = 1/r$. Note that Pollard's Rho is a process with $r = 3$ transition types. We use the additive walk on exponents in our analysis.

The goal of this section is to determine the expected time of the first collision between these processes:

$$\mathbb{E}\min\{i : \exists j < i,\, X_i = X_j\}$$

The non-rigorous part of the heuristic in Section 1.2 is in treating every X_i as if it were an independent uniform random sample from G, and so $p = \mathsf{P}(X_i = X_j) = 1/|G|$ when $i \neq j$, and there are an average $\sqrt{\frac{\pi}{2}p^{-1}}$ samples until collision. We replace this by a claim that every S_i and S_j are pairwise independent, with $p = \mathsf{P}(S_i \cap S_j \neq \emptyset) \approx 1/A$ for some A, and so the expected number of segments until a collision is around $\sqrt{\frac{\pi}{2}\,p^{-1}} = \sqrt{\frac{\pi}{2}\,A}$. The expected run time of the Rho method is then $(L+\tau)\sqrt{\frac{\pi}{2}\,A}$.

This ignores potential collisions involving an R_i or an S_i with itself. However, if $\tau = o(L)$ then collisions involving an R_i segment make up only an $o(1)$ fraction of potential collisions. Likewise, collisions within an S_i are only an $o(1)$ fraction of potential collisions if $L = o(\sqrt{N})$. For the sake of rigor we show in the full version of the paper that these in fact have only a negligible probability of being the location of the first collision.

What is the appropriate value for τ? The re-randomization portion R_i is intended to make S_i independent of earlier segments. It suffices that the first state in S_i be an independent nearly uniform random sample.

$$\forall v, w \in V : \frac{1 - 1/N}{N} \leq \mathsf{P}^{\tau}(v, w) \leq \frac{1 + 1/N}{N}\,.$$

The minimum value of τ for which this holds is called "L^{∞} mixing time." Montenegro, Kim, and Tetali [9] showed that for Pollard's Rho walk $\tau = O(\log^3 N)$, while Hildebrand [8,18] showed that for Teske's additive walk $\tau = O^*(N^{2/(r-1)})$. A slightly weaker notion of mixing should be used for Teske's process when $r < 6$, but we do not consider it here.

We now turn to the proof of Theorem 3. This uses a generalization of the birthday problem. Consider a family of events E_1, E_2, … such that $\mathsf{P}(E_1) = 0$ and $\mathsf{P}(E_k \mid \neg E_{k-1}) = (1 + o(1))\frac{k-1}{A}$. Then

$$\mathbb{E}\min\{t : E_t\} = (1 + O(1/A))\sqrt{\frac{\pi}{2}\,A}\,.$$

We prove a more general form of this in the Appendix.

Let CS_t denote the number of collisions between the first t segments S_1, S_2, ..., S_t. We will take E_t as the event that $\mathrm{CS}_t > 0$, and so we need to determine $\mathsf{P}\left(E_t \mid \neg E_{t-1}\right)$.

Consider segment S_t. It starts at $X_{(t-1)L+t\tau}$, which is a sample within $\epsilon = 1/N$ of uniform, independent of earlier rounds. It proceeds as a random path containing L states. Assume for now that all transitions are equally likely, so that all paths are equally likely as well; we discuss the non-uniform case at the end of the section. There are $N \times r^{L-1}$ possible paths, and each will have probability between $\frac{1\pm 1/N}{Nr^{L-1}}$. In order to have $\mathrm{CS}_t > 0$ the path must collide with one of the $\leq (t-1)L$ points appearing in $S_1, S_2, \ldots, S_{t-1}$, denote this as X_j. There are L positions in the path at which the collision could occur, denote the chosen location as X_i, and r^{L-1} possibilities for the remainder of the path, so at most $(t-1)L^2r^{L-1}$ potential S_t segments include collisions. It follows that

$$\mathsf{P}\left(\mathrm{CS}_t > 0 \mid \mathrm{CS}_{t-1} = 0\right) \leq \frac{(t-1)L^2r^{L-1}(1+1/N)}{Nr^{L-1}(1-1/N)} = \frac{(t-1)(1+O(1/N))}{N/L^2}$$

$$\Rightarrow\ \mathbb{E}\min\{t : \mathrm{CS}_t > 0\} \geq (1+O(L^2/N))\sqrt{\frac{\pi}{2}\frac{N}{L^2}}$$

Each round introduced $\tau + L$ new states, so this suggests run time of

$$\begin{aligned}\mathbb{E}\min\{i : \exists j < i,\, X_i = X_j\} &\gtrsim (\tau+L)(1+O(L^2/N))\sqrt{\frac{\pi}{2}\frac{N}{L^2}} \\ &= (1+O(L^2/N)+O(\tau/L))\sqrt{\frac{\pi}{2}N}\end{aligned}$$

This is just the birthday heuristic. We can improve on this by reducing double-counting of paths. In the construction just given, once X_i and X_j have been decided on, do not construct paths with $X_{i-1} = X_{j-1}$, as these paths will be counted anyway. This reduces the number of segments with $X_i = X_j$ from r^{L-1} to $(r-1)r^{L-2}$, unless X_i or X_j is the first state in their respective segment. This results in:

$$\begin{aligned}\mathsf{P}\left(\mathrm{CS}_t > 0 \mid \mathrm{CS}_{t-1} = 0\right) &\leq \frac{(1-1/L)^2(1-1/r)(t-1)\,L^2\,r^{L-1}}{N\,r^{L-1}}\frac{1+1/N}{1-1/N} \\ &= (1+O(1/L)+O(1/N))\frac{(t-1)(1-1/r)L^2}{N}\end{aligned}$$

$$\Rightarrow\ \mathbb{E}\min\{t : \mathrm{CS}_t > 0\} \geq (1+O(L^2/N))\sqrt{\frac{\pi}{2}\frac{N}{L^2(1-1/r)}}$$

$$\Rightarrow\ \mathbb{E}\min\{i : \exists j < i,\, X_i = X_j\} \gtrsim (1+O(L^2/N)+O(\tau/L))\sqrt{\frac{\pi}{2}\frac{N}{1-1/r}}$$

In the full version of the paper we make the final line rigorous, with the added condition that $j \leq i - \tau$, and so $\gtrsim$ can (almost) be replaced by $\geq$.

The bound can be made more-or-less sharp, but at the cost of rigor. A path with first collision at $X_i = X_j$ will have an average of $\mathcal{C}_\tau$ collisions in the next τ steps, and so we counted colliding paths $\mathcal{C}_\tau$ times each on average. The number of colliding paths is only $(1-\tau/L)/\mathcal{C}_\tau$ of our original rough estimate.

$$\begin{aligned} \mathsf{P}\left(\mathrm{CS}_t > 0 \mid I_t = 0\right) &\approx \frac{(t-1)\,L^2\,r^{L-1}/\mathcal{C}_\tau}{N\,r^{L-1}}\,\frac{1+O(1/N)}{1-O(1/N)} \\ &= \frac{(t-1)\,L^2}{\mathcal{C}_\tau N}\,(1+O(1/N)) \\ \Rightarrow\ \mathbb{E}\min\{t:\,\mathrm{CS}_t>0\} &\approx (1+O(L^2/N))\sqrt{\frac{\pi}{2}\,\frac{N\,\mathcal{C}_\tau}{L^2}} \\ \Rightarrow\ \mathbb{E}\min\{i:\,\exists j<i,\,X_i=X_j\} &\approx (1+O(L^2/N)+O(\tau/L))\sqrt{\frac{\pi}{2}\,N\,\mathcal{C}_\tau} \end{aligned}$$

The non-rigor here is in ignoring the effects of the condition $\mathrm{CS}_{t-1} = 0$ on the expected number of collisions after $X_i = X_j$. However, given that only an $O(1/\sqrt{N})$ fraction of states will be covered before the expected collision time this effect should be quite minimal. Indeed, the simulations data discussed in Section 7 show that our heuristic has 4 or more digits of accuracy.

The approximation $\mathcal{C}_\tau \geq \mathcal{C}_1$ gives our earlier weaker, but rigorous, result, so this is an extension of what we know to be true. Pollard's walk has $\tau = O(\log^3 N)$ [9] and so when $L = \sqrt[3]{N}$ we get run time $(1+O^*(1/\sqrt[3]{N}))\sqrt{\frac{\pi}{2}\,N\,\mathcal{C}_\tau}$. Teske's additive walks have $\tau = O(N^{2/(r-1)})$ [18] and so if $r \geq 6$ then $L = N^{0.5-\epsilon}$ for small ϵ will suffice to show run time of $(1+o(1))\sqrt{\frac{\pi}{2}\,N\,\mathcal{C}_\tau}$, while if $r = 5$ then $L = \sqrt{N}$ will show $O\left(\sqrt{\frac{\pi}{2}\,N\,\mathcal{C}_\tau}\right)$. These are consistent with Teske's observation that the walk slows considerably when $r \leq 4$.

Remark 7. When the transitions have non-uniform probabilities nearly everything just argued still applies, because in our construction of colliding paths we allow all possible transitions to occur. The sole exception is the $1-1/r$ correction. In this case it suffices to replace $1/r$ by the smallest transition probability. That is of course pessimistic, but is difficult to avoid in a rigorous argument. The $\mathcal{C}_\tau$ bound does not suffer this weakness and again seems to be sharp.

5 Gaudry-Schost

There are many variations of the Gaudry-Schost method, with versions to solve multi-dimensional discrete logarithms, to speed up the algorithm by making the regions non-hypercubes, to speed up by considering collision of 3 or 4 walks on differing regions, etc [5–7]. Our technique can be applied in each of these settings, but for simplicity we will consider only the simplest case, an early one-dimensional version [7]. We comment on a few other versions at the end of this section and in Section 7.

The non-rigorous part of the heuristic is in ignoring the dependence between states such as X_i and X_{i+1}. We resolve this by breaking the X and Y walks into segments S_i^X and S_j^X that are independent. The most natural choice for segments is to let S_i^X denote the states visited by walk X between the $(i-1)^{st}$ and i^{th} distinguished points, including the i^{th}, and define S_i^Y and R_i^Y similarly. This implicitly sets $R_i^X = \emptyset$. The length $|S_i|$ is a random variable with geometric distribution $\mathsf{P}\left(|S_i| = \ell\right) = (1-\theta)^{\ell-1}\,\theta$ and expectation $\mathbb{E}|S_i| = \theta^{-1}$.

As before, we consider the probability that two segments collide, this time segments S_i^X and S_j^Y. To simplify the discussion we ignore boundary effects and assume that all segments are in $A \cap B$, since this is the only area in which collisions can occur. There are two main types of boundary effects: when walk X crosses into B it effectively increases the size of $A \cap B$, which improves the runtime, but when it crosses into A^c it effectively increases the size of A which decreases the runtime. A careful analysis finds that these effects almost exactly cancel out.

Let E_t denote the number of times that one of the first t segments for X intersects one of the first t segments for Y. As in the analysis of the Kangaroo method, we will use the relation that for a non-negative random variable $\mathsf{P}\left(Q > 0\right) = \frac{\mathbb{E}Q}{\mathbb{E}(Q \mid Q>0)}$.

First, consider the chance that S_t^X intersects with one of the first $(t-1)$ segments of the Y walk.

$$\begin{aligned}\mathsf{P}\left(S_t^X \cap \left(\cup_{j=1}^{t-1} S_j^Y\right) \neq \emptyset \;\mid\; E_{t-1} = 0\right) &\approx \frac{\theta^{-1} \times (t-1)\,\frac{\theta^{-1}}{|A\cap B|}}{\mathcal{C}_\tau} \\ &= \frac{t-1}{\theta^2 \mathcal{C}_\tau\,|A \cap B|}\end{aligned}$$

Next, consider the chance that S_t^Y intersects with one of the first t segments of the X walk, if it has not collided already:

$$\begin{aligned}\mathsf{P}\left(S_t^Y \cap \left(\cup_{j=1}^{t} S_j^X\right) \neq \emptyset \;\mid\; E_{t-1} = 0 \,\wedge\, S_t^X \cap \left(\cup_{j=1}^{t-1} S_j^Y\right) = \emptyset\right) &\approx \frac{\theta^{-1} \times t\,\frac{\theta^{-1}}{|A\cap B|}}{\mathcal{C}_\tau} \\ &= \frac{t}{\theta^2 \mathcal{C}_\tau\,|A \cap B|}\end{aligned}$$

Then

$$\mathsf{P}\left(E_t > 0 \mid E_{t-1} = 0\right) \approx \frac{t-1}{\theta^2 \mathcal{C}_\tau\,|A\cap B|} + \left(1 - \frac{t-1}{\theta^2 \mathcal{C}_\tau\,|A\cap B|}\right)\frac{t}{\theta^2 \mathcal{C}_\tau\,|A\cap B|} \tag{3}$$

The two-walk birthday problem generalizes to say that if E_t is a non-negative random variable such that

$$\mathsf{P}\left(E_t > 0 \mid E_{t-1} = 0\right) = \frac{t-1}{\mathcal{N}} + \left(1 - \frac{t-1}{\mathcal{N}}\right)\frac{t}{\mathcal{N}}$$

then

$$\mathbb{E}\min\{t:\, E_t > 0\} = (1 + O(1/\mathcal{N}))\,\frac{1}{2}\sqrt{\pi\mathcal{N}}\,.$$

Equation (3) satisfies the condition when $\mathcal{N} = \theta^2 \mathcal{C}_\tau \, |A \cap B|$, and so each walk requires an average of $\frac{1}{2}\sqrt{\pi \mathcal{N}}$ segments in $A \cap B$. If we ignore boundary effects, then each segment from the X walk has probability $\frac{|A \cap B|}{|A|}$ of being in $A \cap B$, while each segment from Y has probability $\frac{|A \cap B|}{|B|}$ of this. So drawing the required number of samples from $A \cap B$ requires an average of

$$\frac{|A|}{|A \cap B|} \, \frac{1}{2} \, \sqrt{\pi \theta^2 \, \mathcal{C}_\tau \, |A \cap B|}$$

segments from X and a similar number from the Y process. Each segment involved an average of θ^{-1} steps of the walk, so the number of steps of the walks is

$$\frac{|A| + |B|}{2|A \cap B|} \, \sqrt{\pi \, \mathcal{C}_\tau \, |A \cap B|} \, .$$

For instance, Galbraith and Ruprai's improved version of Gaudry-Schost [6] uses regions with $|A|/|A \cap B| = |B|/|A \cap B| = 2$ and $|A \cap B| = N/3$. This leads to a runtime estimate of $\frac{2}{3^{1/2}} \sqrt{\pi \, \mathcal{C}_\tau \, N}$, which is a factor $\sqrt{\mathcal{C}_\tau}$ times slower than previous predictions. Our simulations find that this is within 0.3% of the correct runtime. See Section 7 for further details.

6 The Collision Number

Almost all of our bounds consider the collision number $\mathcal{C}_\tau$. We remind the reader of its definition.

Definition 8. *The* collision number $\mathcal{C}_\tau$ *is the expected number of collisions when two independent copies of a random walk start at the same state, chosen uniformly at random, and proceed for τ steps.*

The fact that this is an average case behavior means that we can ignore the possibility of bad start values, as these are rare. Determining this value exactly is still generally prohibitive, but upper and lower bounds of arbitrary precision are possible.

The simplest approximation on the collision number is the lower bound $\mathcal{C}_\tau \geq \mathcal{C}_\ell$ for $\ell \leq \tau$. When $\mathcal{C}_\tau \geq \mathcal{C}_0 = 1$ is used our bounds simply reduce to the heuristic results of Section 1.2. When $\mathcal{C}_\tau \geq \mathcal{C}_1 = 1 + 1/r$ is used the Rho heuristic of Heuristic 4 reduces to Theorem 3; this also gives Blackburn and Murphy's heuristic. When $\mathcal{C}_\tau \geq \mathcal{C}_2$ is used then we start producing new results.

For small terms such as $\mathcal{C}_2$ it is typically possible to compute the value exactly by hand. We give a few examples below.

Another method of estimating $\mathcal{C}_\tau$ is to observe that most collisions will occur quickly, and once a collision does occur then it should be followed by roughly another $\mathcal{C}_\tau$ collisions. Hence, if p_ℓ is the probability of a collision within a small number of steps ℓ then

$$\begin{aligned} & \mathcal{C}_\tau \geq 1 + p_\ell \, \mathcal{C}_\tau \\ \Rightarrow \; & \mathcal{C}_\tau \geq \frac{1}{1 - p_\ell} \end{aligned} \tag{4}$$

For very small ℓ this can be computed by hand, but it is usually better to involve a computer. The values produced by this estimate are quite accurate.

Simulation data shown in Section 7 shows that our heuristics are very precise. The methods of computing each $\mathcal{C}_\tau$ will not differ much, so we only give detailed work for Pollard's Rho while keeping the work short for Pollard's Kangaroo and Gaudry-Schost.

Example 9 (Pollard's Kangaroo). Consider p_1. This requires both walks to make the same initial transition $X_1 = X_0 + 2^k = Y_1$, so $p_1 = 1/(d+1)$ and

$$\mathcal{C}_\tau \geq \frac{1}{1-p_1} = 1 + \frac{1}{d} = 1 + \frac{2}{\log_2 N}$$

Consider p_2. This requires both to make the same initial transition, or do the first two steps in reversed order, or one walk does the same step twice making it add up to the other walk's value. Then

$$\begin{aligned} p_2 &= \frac{1}{d+1} + \binom{d+1}{2} \frac{1}{(d+1)^4} + 2d\,\frac{1}{(d+1)^3} \\ &= \frac{2}{\log_2 N} + \frac{10}{(\log_2 N)^2} + O((\log N)^{-3})\,. \end{aligned}$$

As $N \to \infty$ this goes to zero, so we can use the relation $1/(1-p) \to 1 + p + p^2 + \cdots$. This gives the relation

$$\mathcal{C}_\tau \geq 1 + \frac{2}{\log_2 N} + \frac{14}{(\log_2 N)^2} + O((\log N)^{-3})\,.$$

This is quite accurate. See Section 7 for a plot using a version of this bound.

Example 10 (Teske's Additive Walks). For Teske's additive version of Pollard's Rho there are r step types of the form $X \to X + s_i$. Consider the probability that two independent walks with $X_0 = Y_0$ intersect within $\ell = 3$ steps:

$$\begin{aligned} &\mathsf{P}\left(\exists i, j \leq 3,\ X_i = Y_j \mid X_0 = Y_0\right) \\ &= \mathsf{P}\left(X_1 = Y_1 \mid X_0 = Y_0\right) + \mathsf{P}\left(X_2 = Y_2,\ X_1 \neq Y_1 \mid X_0 = Y_0\right) \\ &\quad + \mathsf{P}\left(X_3 = Y_3,\ X_2 \neq Y_2,\ X_1 \neq Y_1 \mid X_0 = Y_0\right) \\ &= \mathsf{P}\left(X_1 = X_0 + s_i = Y_0 + s_i = Y_1\right) \\ &\quad + \mathsf{P}\left(X_2 = X_0 + s_i + s_j = Y_0 + s_j + s_i = Y_2,\ i \neq j \mid X_0 = Y_0\right) \\ &\quad + \mathsf{P}\left(X_3 = X_0 + s_i + s_j + s_k = Y_0 + s_j + s_k + s_i,\ i \neq j \neq k \neq i, \mid X_0 = Y_0\right) \\ &= \frac{1}{r} + {}_r\mathrm{P}_2 \frac{1}{r^2}\frac{1}{r^2} + \frac{3\,{}_r\mathrm{P}_3 + 2\,{}_r\mathrm{P}_2}{r^6} \\ &= \frac{1}{r} + \frac{1}{r^2} + \frac{2}{r^3} + O(1/r^4) \end{aligned}$$

It follows that

$$\mathcal{C}_\tau \approx \frac{1}{1-p_3} = \frac{1}{1 - \left(\frac{1}{r} + \frac{1}{r^2} + \frac{2}{r^3} + O(1/r^4)\right)} = 1 + \frac{1}{r} + \frac{2}{r^2} + \frac{4}{r^3} + O(1/r^4)$$

See Section 7 for discussion of the accuracy of this.

Example 11 (Gaudry-Schost). The step types were chosen uniformly at random from an interval, and so with high probability a collision will occur in a short number of steps iff the same steps are taken by both walks, or the same steps are taken but with the order re-arranged. This is just what happens with Teske's additive walks, and so we may borrow the work done when we examined her methods. Namely,

$$\mathcal{C}_\tau \approx \frac{1}{1-p_3} = \frac{1}{1-\left(\frac{1}{r}+\frac{1}{r^2}+\frac{2}{r^3}+O(1/r^4)\right)} = 1+\frac{1}{r}+\frac{2}{r^2}+\frac{4}{r^3}+O(1/r^4)$$

7 Sharpness of our Results

We have consolidated our simulation details here. All of these show that our results are extremely precise.

We note that our simulations are done in a non-standard way. Our goal is to study performance of various methods for finding the discrete logarithm, not to study the strengths or weaknesses of specific hash functions or representations of a cyclic group. As a result we study walks on the exponents, not the group. For instance, the walk $X_0 = h = g^x$, $X_1 = h\,g$, $X_2 = (h\,g)^2$ is equivalent to $x \to x+1 \to 2x+2 \mod |G|$. The hash used to do a walk on the exponent was based on the Mersenne Twister [11], as it is a fast source of pseudo-randomness. Several variations on this hash were tested, and it was confirmed that run time was similar in each case.

We first consider Pollard's Kangaroo with power of two steps.

Example 12 (Pollard's Kangaroo). When $N = |G| = 10^9$ Figure 1 shows that there is still a significant gap between simulation data and prior heuristics, but that our new result almost exactly matches the simulation results.

Pollard's degree 3 process is a useful test cases as its performance deviates from simple heuristic much more than does Teske's improved process.

Example 13 (Pollard's Rho). Very large simulations show that $\mathcal{C}_\tau = 1.68221 \pm 0.00001$. A computer can be used to enumerate all possible paths of length $\ell \le 20$. This gives the estimate $\mathcal{C}_\tau \approx \frac{1}{1-p_5} = 1.65237$, while $\mathcal{C}_\tau \approx \frac{1}{1-p_{10}} = 1.67730$, and $\mathcal{C}_\tau \approx \frac{1}{1-p_{20}} = 1.68203$. So even p_{10} was sufficient to give an estimate of $\mathcal{C}_\tau$ within 0.3% of the true value.

This can be seen more clearly visually. Figure 2 shows simulation data for runtime and finds that it is consistently around $1.6254\sqrt{N}$. Figure 3 shows that $\mathcal{C}_t$ approaches $\mathcal{C}_\tau$ fairly quickly in t, with $\mathcal{C}_{20} \approx \mathcal{C}_\tau$, and leads to a runtime prediction of $1.6256\sqrt{N}$.

We next consider Teske's r-adding version of the Rho method.

Example 14 (Teske's additive walks).

Teske estimates average collision time of the 20-adding walk is around $1.292\sqrt{N}$ steps [18]. We did a much larger run of 75 million simulations and

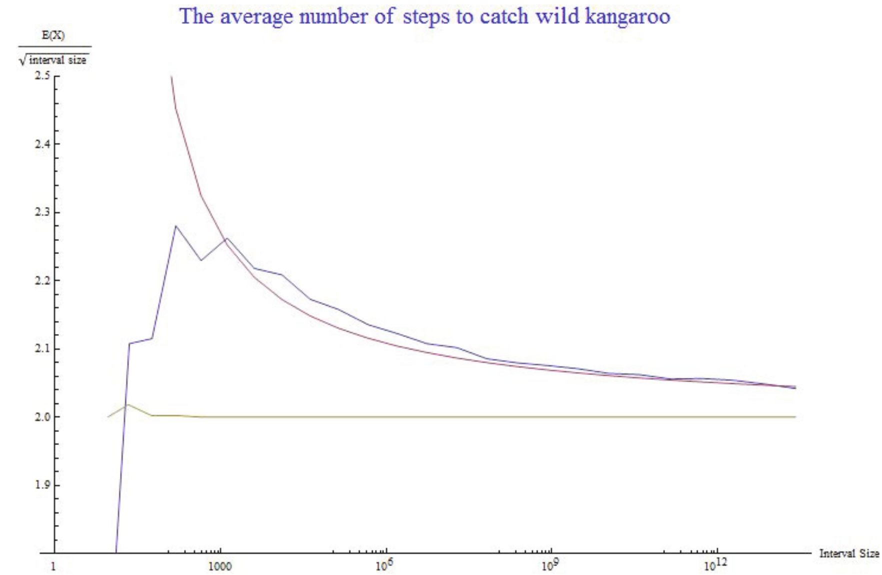

Fig. 1. Standard heuristic (flat line), our bound (smooth curve), simulation data with margin of error $\pm 0.01\sqrt{N}$ (jagged plot)

found a 95% confidence interval of $1.2877\sqrt{N}$ to $1.2880\sqrt{N}$. This suggests that collision time is about 3% slower than the $1.2533\sqrt{N}$ steps predicted by the birthday heuristic.

To apply our heuristic, recall from Section 6 that

$$\mathcal{C}_\tau \approx \frac{1}{1-p_3} = \frac{1}{1-\left(\frac{1}{r}+\frac{1}{r^2}+\frac{2}{r^3}+O(1/r^4)\right)}$$

When $r = 20$ this leads to an estimate on collision time of $1.2877\sqrt{N}$, which is already within the 95% confidence interval given by simulation data. An exact enumeration of walks of length $\ell = 5$ increases the estimate only negligibly to $1.287765\sqrt{N}$ steps, at $\ell = 10$ to $1.287770\sqrt{N}$ steps, and the sampling based estimate at length $\ell = 100$ gave an estimate of $(1.287769 \pm 0.000003)\sqrt{N}$ with 95% confidence.

So in this case a mere 3 steps already explains 99.7% of the 20-additive walk's deviation from the birthday heuristic, and by 5 steps the estimate is essentially sharp.

Last of all, we compare simulation data to our heuristic for Gaudry-Schost.

Example 15. Galbraith, Pollard, and Rubrai [5] discuss 3 and 4-walk versions with even better runtime than Pollard's Kangaroo method. The same argument used to give a heuristic bound for Gaudry-Schost shows that this will have a $\sqrt{\mathcal{C}_\tau}$ slowdown over their predicted runtime. They consider an interval of side

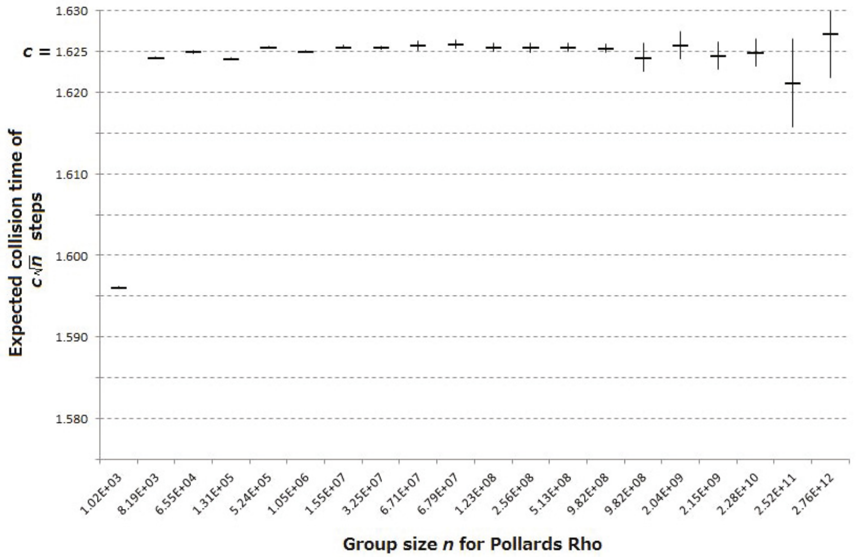

Fig. 2. Run time of Pollard's Rho: Simulations estimate $(1.6254 \pm 0.0004)\sqrt{N}$ steps

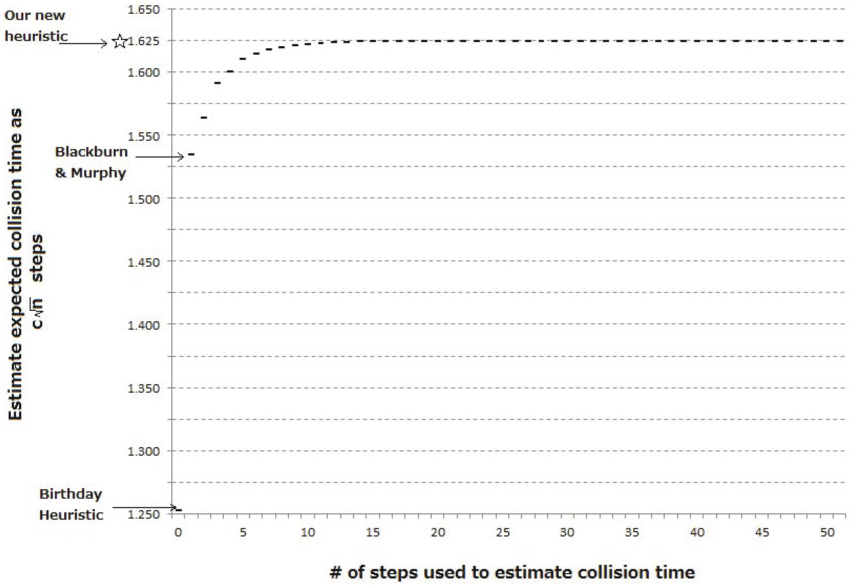

Fig. 3. Run time of Pollard's Rho: Heuristic predicts $1.6256\sqrt{N}$ steps

$N = 2^{40}$, with $\theta = 1/500$, $r = 32$ step types, and average step size $m = 0.01\,\theta\,N$, and determine that a basic birthday heuristic suggests run time of $1.761\sqrt{N}$. Our improved heuristic will be $\approx 1.761/\sqrt{1 - 1/32 - 1/32^2 - 2/32^3} = 1.790\sqrt{N}$. We did 250, 000 runs without any adjustment for boundary effects and found mean run time of $1.79501\,\sqrt{N}$ with 95% CI of $(1.791, 1.799)\,\sqrt{N}$.

Our heuristic suggests that run time will improve when there are more generators, whereas past heuristics said nothing about this. We repeated the above simulations but with $d = 128$ step types. This time the average run time was $1.77113\sqrt{N}$ with 95% CI of $(1.767, 1.775)\,\sqrt{N}$. Our heuristic of $1.768\sqrt{N}$ is within this interval. Using more generators does indeed help, and the improvement is predicted fairly accurately by our heuristic. The error is again about 0.3% from the center of the interval.

This shows that our improved heuristic is fairly good. The error in each case is near the bottom of the CI, and only 0.3% from the center of the interval. Presumably any error is due to minor boundary effects.

Example 16. One method that has been proposed for avoiding boundary effects is to forbid starts in the rightmost $0.01\,N$ of an interval, as this is the average distance traveled before a distinguished point is reached. We tested this in the case above, with $r = 32$, and found mean run time of $1.79721\sqrt{N}$ with 95% CI of $(1.793, 1.801)\sqrt{N}$. This is not a statistically significant difference from the case that ignores boundary issues.

A Appendix

When looking at the Rho algorithm we required a generalization of the birthday problem. We prove that here.

Consider a family of events $E_1 \subseteq E_2 \subseteq \cdots$ such that $\mathsf{P}\left(E_k \mid \neg E_{k-1}\right) \leq \frac{k-1}{A}$, and a second family $F_1 \subseteq F_2 \subseteq \cdots$. We will modify the birthday result in order to prove a result about the expected time until some event is true.

$$\mathbb{E}\min\{t:\ E_t \cup F_t\} = \sum_{t=0}^{\infty} \mathsf{P}\left(\neg(E_t \cup F_t)\right)$$

In our case, E_t will be the event that a collision has occurred between segments S_1, S_2, …, S_t, as was considered earlier in the paper, while F_t will be the event that a collision occurred elsewhere: within one of the segments S_1, S_2, …, S_t or involving one of the randomization segments R_1, R_2, …, R_t. The collision time will be $(L+T)\mathbb{E}\min\{t:\ E_t \cup F_t\}$.

First consider collisions between segments.

$$\begin{aligned}\mathsf{P}\left(\neg E_t\right) &= \mathsf{P}\left(\neg E_1\right)\prod_{k=2}^{t}\mathsf{P}\left(\neg E_k \mid \neg E_{k-1}\right)\\ &\geq 1\prod_{k=2}^{t}(1-\frac{k-1}{A})\end{aligned}$$

This is exactly the probability that occurs in the birthday problem when there are A days in the year, and so

$$\mathbb{E}\min\{t:\ E_t\} = \sum_{t=0}^{\infty} \mathsf{P}\left(\neg E_t\right) \geq (1+O(1/A))\sqrt{\frac{\pi}{2}\,A} \tag{5}$$

It follows that for any value of T

$$\begin{aligned}\mathbb{E}\min\{t:\ E_t \cup F_t\} &= \sum_{t=0}^{\infty} \mathsf{P}\left(\neg(E_t \cup F_t)\right)\\ &\geq \sum_{t=0}^{T-1} \mathsf{P}\left(\neg E_t\right) - \sum_{t=0}^{T-1} \mathsf{P}\left(F_t\right)\\ &\geq (1+O(1/A))\sqrt{\frac{\pi}{2}\,A} - \sum_{t=T}^{\infty} 1 \prod_{k=2}^{t}(1-\frac{k-1}{A}) - \sum_{t=0}^{T-1} \mathsf{P}\left(F_t\right)\end{aligned}$$

The tail probability in the first sum can be estimated as

$$\begin{aligned}\sum_{t=T}^{\infty} 1 \prod_{k=2}^{t}(1-\frac{k-1}{A}) &\leq \sum_{t=T}^{\infty} \exp\left(-\sum_{k=2}^{t}\frac{k-1}{A}\right)\\ &= \sum_{t=T}^{\infty} \exp\left(-t(t-1)/2A\right)\\ &\leq \sum_{t=T}^{\infty} \int_{t-2}^{t-1} e^{-x^2/2A}\,dx\\ &= \int_{T-2}^{\infty} e^{-x^2/2A}\,dx\\ &\leq \int_{(T-2)/\sqrt{A}}^{\infty} \frac{u}{(T-2)/\sqrt{A}}\,e^{-u^2/2}\,(\sqrt{A}\,du)\\ &= \frac{A}{T-2}\,e^{-(T-2)^2/2A}\end{aligned}$$

The final inequality involved the substitution $u = x/\sqrt{A}$ and the relation $u \geq \frac{T-2}{\sqrt{A}}$.. When $T \geq 2+\sqrt{A\log A}$ then this is $o(1)$. Then

$$\mathbb{E}\min\{t:\ E_t \cup F_t\} \geq (1+O(1/A))\sqrt{\frac{\pi}{2}\,A} - \frac{A^{3/2}}{T-2}\,e^{-(T-2)^2/2A} - \sum_{t=0}^{T-1} \mathsf{P}\left(F_t\right)\ .$$

We found in Section 4 that $A = \frac{N}{L^2(1-1/r)}$ can be shown rigorously, while $A = \frac{N\,\mathcal{C}_\tau}{L^2}$ can be shown heuristically.

References

1. Bailey, D., Batina, L., Bernstein, D., Birkner, P., Bos, J., Chen, H.-C., Cheng, C.-M., Van Damme, G., de Meulenaer, G., Perez, L.J.D., Fan, J., Güneysu, T., Gürkaynak, F., Kleinjung, T., Lange, T., Mentens, N., Niederhagen, R., Paar, C., Regazzoni, F., Schwabe, P., Uhsade, L., Van Herrewege, A., Yang, B-Y.: "Breaking ECC2K-130," Cryptology ePrint Archive, Report 2009/541 (2009). https://eprint.iacr.org/2009/541
2. Bernstein, D.J., Lange, T.: Two grumpy giants and a baby. In: ANTS X: Proceedings of the 10th International Symposium on Algorithmic Number Theory. Mathematical Sciences Publishers (2013)
3. Blackburn, S., Scott, S.: The discrete logarithm problem for exponents of bounded height. In: ANTS XI: Proceedings of the 11th International Symposium on Algorithmic Number Theory. LMS J. Comput. Math **17**, 148–156 (2014)
4. Blackburn, S., Murphy, S.: The number of partitions in Pollard Rho, Unpublished note : Later made available as Technical report RHUL-MA-2011-11 (Department of Mathematics, p. 2011. University of London, Royal Holloway (1998)
5. Galbraith, S.D., Pollard, J.M., Ruprai, R.S.: Computing discrete logarithms in an interval. Math. Comp. **82**, 1181–1195 (2013)
6. Galbraith, S., Ruprai, R.S.: An improvement to the Gaudry-Schost algorithm for multidimensional discrete logarithm problems. In: Parker, M.G. (ed.) Cryptography and Coding 2009. LNCS, vol. 5921, pp. 368–382. Springer, Heidelberg (2009)
7. Gaudry, P., Schost, É.: A low-memory parallel version of Matsuo, Chao, and Tsujii's algorithm. In: Buell, D.A. (ed.) ANTS 2004. LNCS, vol. 3076, pp. 208–222. Springer, Heidelberg (2004)
8. Hildebrand, M.: On the Chung-Diaconis-Graham random process. Electron. Comm. Probab. **11**, 347–356 (2006)
9. Kim, J-H., Montenegro, R., Tetali, P.: Near Optimal Bounds for Collision in Pollard Rho for Discrete Log. In: IEEE Proc. of the Symposium on Foundations of Computer Science (FOCS 2007), pp. 215–223 (2007)
10. Kim, J.-H., Montenegro, R., Peres, Y., Tetali, P.: A Birthday Paradox for Markov chains, with an optimal bound for collision in the Pollard Rho Algorithm for Discrete Logarithm. The Annals of Applied Probability **20**(2), 495–521 (2010)
11. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation **8**(1), 3–30 (1998)
12. Miller, S.D., Venkatesan, R.: Spectral analysis of Pollard rho collisions. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 573–581. Springer, Heidelberg (2006)
13. Montenegro, R., Tetali, P.: How long does it take to catch a wild kangaroo?. In: Proc. of 41st ACM Symposium on Theory of Computing (STOC 2009), pp. 553–559 (2009). Citations refer to an improved version at http://arxiv.org/pdf/0812.0789v2.pdf
14. Nishimura, K., Shibuya, M.: Probability to meet in the middle. Journal of Cryptology **2**(1), 13–22 (1990)
15. Pollard, J.: Monte Carlo methods for index computation mod p. Mathematics of Computation **32**(143), 918–924 (1978)
16. Pollard, J.: Kangaroos, Monopoly and Discrete Logarithms. Journal of Cryptology **13**(4), 437–447 (2000)

17. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
18. Teske, E.: Speeding up Pollard's rho method for computing discrete logarithms. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 541–554. Springer, Heidelberg (1998)
19. Rosini, M.D.: Applications. In: Rosini, M.D. (ed.) Macroscopic Models for Vehicular Flows and Crowd Dynamics: Theory and Applications. UCS, vol. 12, pp. 217–226. Springer, Heidelberg (2013)