J. Math. Cryptol. 8 (2014), 71–92

Collision bounds for the additive Pollard rho algorithm for solving discrete logarithms

Joppe W. Bos, Alina Dudeanu and Dimitar Jetchev

Communicated by Rainer Steinwandt

Abstract. We prove collision bounds for the Pollard rho algorithm to solve the discrete logarithm problem in a general cyclic group $\mathbf{G}$. Unlike the setting studied by Kim et al., we consider additive walks: the setting used in practice to solve the elliptic curve discrete logarithm problem. Our bounds differ from the birthday bound $\mathcal{O}(\sqrt{|\mathbf{G}|})$ by a factor of $\sqrt{\log|\mathbf{G}|}$ and are based on mixing time estimates for random walks on finite abelian groups due to Dou and Hildebrand.

Keywords. Pollard rho, additive walk, collision bound, random walk, mixing times.

2010 Mathematics Subject Classification. 94A60, 05C81.

1 Introduction

Let $\mathbf{G}$ be a finite cyclic group of prime order and let $g \in \mathbf{G}$ be a generator. Given g and an element $h \in \mathbf{G}$, the *discrete logarithm problem* (DLP) is the problem of computing an integer y such that $h = g^y$. This problem is *believed* to be hard in the case for certain groups of points on elliptic curves over a finite field but can be solved in subexponential time for multiplicative groups of finite fields [1] and for Jacobians of hyperelliptic curves of high genus [2,3,10,12,14,18]. For this reason, the *elliptic curve discrete logarithm problem* (ECDLP) is used as the theoretical foundation of many standardized protocols used in elliptic curve cryptography [25, 28]. To highlight the practical importance of this problem, the United States' National Security Agency restricts the use of public key cryptography in "Suite B" [31] to ECC only.

The parallelized [37] Pollard rho algorithm [33] is one of the most commonly used methods for solving the discrete logarithm problem when $\mathbf{G}$ is a generic finite cyclic group. The basic idea is to define a *walk* over the elements of the group $\mathbf{G}$ using an iteration function. One might solve the DLP when a group element is

This work was supported by the Swiss National Science Foundation under grant number 200020-132160.

encountered twice (such an event is commonly referred to as a *collision*). If one assumes that the elements from the walk generated by this iteration function are independent and uniformly random among all elements of $\mathbf{G}$, the birthday paradox implies that one can obtain a collision with probability greater than 50% after $\mathcal{O}(\sqrt{|\mathbf{G}|})$ steps and with high probability after $\mathcal{O}(\sqrt{|\mathbf{G}|\log|\mathbf{G}|})$ steps [13, 17]. Here, *with high probability* means that the probability of success is $1 - \mathcal{O}(|\mathbf{G}|^{-c})$ for some $c > 0$ that does not depend on $|\mathbf{G}|$.

Obviously, the assumption that the points generated by the Pollard rho iteration function are independent and uniformly random among all elements of $\mathbf{G}$ is incorrect (see Section 2.2 for more details). This has motivated a line of research to *rigorously* prove the desired bound of $\mathcal{O}(\sqrt{|\mathbf{G}|})$ matching the lower bound for solving discrete logarithms for generic algorithms in prime order groups by Shoup [35] (in the black-box model). This is an active area demonstrated by the fact that the run-time of another generic method to solve the DLP, the Pollard kangaroo method [33, 34], has been rigorously proven correct [29]. The rigorous proof for the Pollard rho method was established using Markov chains by Kim, Montenegro, Peres and Tetali in 2008 [21, 22] improving on previous attempts [23, 26].

Unfortunately, the proof presented in [21, 22] only works for the original iteration function used by Pollard in [33]. In practice, however, the so-called additive walks as studied by Teske [36] are used to solve instances of the DLP (see, e.g., the methods described in [6, 9, 16] for solving the discrete logarithm problem for elliptic curves). A property from the original Pollard rho iteration function that is not present in these additive walks is crucial for establishing rapid mixing results for random walks in the proof by Kim et al.

As far as we know, this paper is the first attempt to rigorously prove the run time of the additive Pollard rho method. It is well known from experimental data [7, 36] and heuristic arguments [4, Appendix B] that by increasing the number of partitions in the additive walk, the performance of the iteration function better resembles the behavior of a truly random walk. We use a model introduced by Greenhalgh [15] and extended by Hildebrand [20, Theorem 2] where the number of partitions in the iteration function depends logarithmically on the cardinality of the group $\mathbf{G}$. This is in agreement with the intuition that one should use more partitions when larger instances of the DLP are being solved. Using this idea together with results about estimating mixing times for random walks on additive groups due to Dou and Hildebrand [11, 19], we prove a collision bound of $\mathcal{O}(\sqrt{|\mathbf{G}|\log|\mathbf{G}|})$ with probability greater than 50% and a collision bound of $\mathcal{O}(\sqrt{|\mathbf{G}|\log|\mathbf{G}|})$ with high probability (see Corollary 3.2). Hence, we are short by a factor $\sqrt{\log|\mathbf{G}|}$ from the birthday bound for both the case of 50% probability of success and the case of high probability of success.

The paper is organized as follows: Section 2 states the preliminaries related to Pollard rho and motivates our work. In Section 3, we explain why the recent methods of Kim et al. [21,22] are not applicable in any obvious way to the setting of the additive Pollard rho algorithm. In Section 4, we recall some basics on random walks on groups, convolutions of functions from Fourier analysis and their links to the distributions of end-points of random walks. Section 5 is devoted to the proof of our main theorem.

2 Preliminaries and motivation

2.1 The classical Pollard rho method

Throughout, we use multiplicative notation for the group $\mathbf{G}$ of prime order N. The Pollard rho algorithm, originally proposed as an integer factorization method [32], was later modified to obtain one of the most commonly used methods for solving the *discrete logarithm problem* when $\mathbf{G}$ is a generic finite cyclic group [33]. In this context, *generic* means that the application of the algorithm is independent of the representation of the group elements, i.e., the algorithm works for any representation as long as the group operation and the operation of testing equality of two group elements are both efficient.

The original Pollard rho method works as follows: partition the group $\mathbf{G}$ into three sets S_1, S_2 and S_3 of roughly the same size, pick a random element $v_0 \in [0, |\mathbf{G}| - 1]$, compute $x_0 = g^{v_0} \in \mathbf{G}$ and for $i \geq 0$ let $x_{i+1} = f(x_i)$ where $f : \mathbf{G} \to \mathbf{G}$ is defined as follows:

$$f(x) = \begin{cases} gx & \text{if } x \in S_1, \\ hx & \text{if } x \in S_2, \\ x^2 & \text{if } x \in S_3. \end{cases}$$

The sequence $\{x_i\}_{i \geq 0}$ represents a walk on $\mathbf{G}$ that will eventually enter a cycle, i.e., there will be integers $m > n$ such that $x_m = x_n$ (we say that we have obtained a collision). Since each x_n is of the form $g^{u_n} h^{v_n}$ for some known $u_n, v_n \in \mathbf{Z}$, we obtain $u_m + y v_m \equiv u_n + y v_n \bmod N$, i.e., unless $v_m \equiv v_n \bmod N$, the solution of the discrete logarithm problem is $y = \frac{u_n - u_m}{v_m - v_n} \bmod N$. Using standard cycle-detection algorithms, such as Floyd's cycle finding method [24, Exercise 3.1.6], the above method requires to store a constant number of group elements. If one makes the heuristic assumption that the subsequent elements of the Pollard rho walk are independent and uniformly random, one would get (by using the birthday bound) that it takes $\mathcal{O}(\sqrt{|\mathbf{G}|})$ steps in order to obtain a collision with probability greater than 50%.

2.2 Complexity analysis

Regardless of the simplicity of the above method, a mathematically rigorous runtime analysis is a rather subtle question of probability theory and statistics. There are two separate stages for analyzing the complexity of solving the discrete logarithm problem via Pollard rho: (1) one needs upper bounds for the number of steps required to obtain a collision in the Pollard rho walk; (2) one needs to estimate the probability of the collision being degenerate. In this paper, we only restrict to (1) and only note that (2) has been carried out rigorously for the classical version of the algorithm in [27].

Regarding (1), we start with the following heuristic argument: if one makes the false assumption that the elements $x_0, x_1, \ldots$ (at least up to the first step when a collision is obtained) are independent and uniformly random among all elements of $\mathbf{G}$, the birthday paradox would imply that one can reach a collision with probability greater than 50% after $\mathcal{O}(\sqrt{|\mathbf{G}|})$ steps and with high probability after $\mathcal{O}(\sqrt{|\mathbf{G}|\log|\mathbf{G}|})$ steps [13, 17]. The elements $x_0, x_1, \ldots$ are, however, far from being independent and uniformly random as they are constructed using the random initial point, the iteration function f and the partitioning $\mathbf{G} = S_1 \sqcup S_2 \sqcup S_3$. The obvious goal is to rigorously prove the birthday bound $\mathcal{O}(\sqrt{|\mathbf{G}|})$ that is observed in practice.

The function f in the original Pollard rho algorithm is fixed. As far as choosing the partition $\mathbf{G} = S_1 \sqcup S_2 \sqcup S_3$ is concerned, we can view it as being given by a function $\ell\colon \mathbf{G} \to \{1, 2, 3\}$. The number of steps it takes to obtain a collision depends on the choice of the function ℓ. It is clear that for certain (degenerate) choices, this number can be quite large (e.g., if $S_1 = \mathbf{G}$ and $S_2 = S_3 = \emptyset$ then it can take as many as $|\mathbf{G}|$ steps to obtain a collision). What one could hope for is that for a random choice of ℓ (selected among some prescribed distribution on the set of all such functions), the collision time will be what we expect (namely, $\mathcal{O}(\sqrt{|\mathbf{G}|})$). If we consider the steps of the Pollard rho walk as random variables $X_0, X_1, \ldots$, the collision time T will then be a stopping time random variable where the randomness is determined by the choice of ℓ as well as by the random choice of the initial element. One could then try to show that with high probability, $T = \mathcal{O}(\sqrt{|\mathbf{G}|})$. We refer to this probabilistic model as *Model 1*. It is clear that the values of $X_1, \ldots, X_n, \ldots$ are completely determined by the choices of X_0 and ℓ. Unfortunately, it is not known how to analyze the statistical behavior of Model 1. A common approach to remedy this problem is to use pseudo-random walks in order to approximate (statistically) the random variables X_i with other random variables that are easier to work with. The idea is that X_{n+1} can be modeled as being computed from X_n by using (with probability $1/3$) a random transition out of the three different transition steps. This gives us the model of a random walk on

the so-called Pollard rho graph (a 3-regular graph whose vertices are the elements of $\mathbf{G}$ and whose edges are determined by the transition steps). Of course, once we obtain a collision, the walk should no longer be random, but deterministic. We refer to this approximation model as *Model 2*. In this case (for the classical Pollard rho), one can show that with probability more than 50% (or, more generally, with any probability that is independent of $|\mathbf{G}|$), a collision occurs after $\mathcal{O}(\sqrt{|\mathbf{G}|})$ steps. In fact, this was not known until recently: the desired bound of $\mathcal{O}(\sqrt{|\mathbf{G}|})$ was rigorously established using Markov chains by Kim, Montenegro, Peres and Tetali in 2008 [21, 22] improving on [23, 26]. The argument relies on establishing rapid mixing results for random walks in the Pollard rho graph where the squaring step plays a crucial role (see Section 3.1 for more details). Yet, currently nothing is known about how well Model 2 approximates Model 1 or vice versa.

2.3 Additive Pollard rho method

In practice (e.g., for groups of points on elliptic curves), the squaring step is often avoided due to inefficiency reasons to solve elliptic curve discrete logarithm problems. Instead, a small integer r is chosen and an r-tuple $(s_1, \ldots, s_r)$ of group elements (transition steps) is precomputed. Given a partition function $\ell\colon \mathbf{G} \to \{1, 2, \ldots, r\}$, one uses the transition function $f(x) = x + s_{\ell(x)}$ instead of the function defined in the original Pollard rho method. This gives rise to a variation of Pollard rho that uses no squaring steps. These walks are known as r-adding walks [36].

The theoretical question studied in this paper is relevant as it is the first attempt to provide a rigorous analysis of the variation of Pollard rho that is most commonly used nowadays. In practice, when solving the discrete logarithm problem, one uses a parallel version of Pollard rho [37]. This leads to an m-fold speedup when the workload is shared among m computational units. The main cost of the Pollard rho method is computing the iteration function f. Computing a single step in the Pollard rho walk (a single iteration of f), is equivalent to computing the group operation in $\mathbf{G}$. In the setting of elliptic curves where we use the additive notation for the group operation, this operation is either a point doubling or a point addition.

Montgomery's *simultaneous inversion* method is often used to speed up Pollard rho [30]. When processing m independent walks in the parallel version of the algorithm, the simultaneous inversion method allows one to substitute m inversions by $3m - 3$ multiplications and a single inversion. When used in combination with affine Weierstrass coordinates, this results in a cost (ignoring modular additions and subtractions) of s squarings, $2 + \frac{3m-3}{m}$ multiplications and $\frac{1}{m}$ inversions to implement the group operation for a single walk where $s = 1$ and $s = 2$ for elliptic curve addition and point doubling, respectively. This makes affine Weier-

strass coordinates the preferred point representation for this type of application and shows that from a performance perspective, point doublings are to be avoided.

3 Run-time analysis for r-adding walks

Analyzing Model 1 is out of reach even in the setting of the classical Pollard rho method. We thus restrict ourselves to Model 2 in the setting of the Pollard rho method using r-adding walks.

3.1 Classical Pollard rho and block walks

The method of Kim, Montenegro, Peres and Tetali [21,22] applies to the classical version of Pollard rho that uses three transition steps, one of which is the squaring step and the other two are multiplications by g and h, respectively. Under Model 2, this is achieved by choosing transition steps uniformly at random among the three until a collision has been obtained. Under this model, it is shown that a collision is obtained in $\mathcal{O}(\sqrt{|\mathbf{G}|})$ steps. The key observation is the fact that one can split the pseudo-random walk into blocks using the squaring steps of the walk as a separating move. More precisely, if one represents the walk by the random variables $X_i = g^{Y_i}$ for unique Y_i's in $[0, |\mathbf{G}|-1]$ then one defines a new sequence of random variables $\{T_i\}$ as follows: let $T_0 = 1$ and let T_1 be the first step when the walk makes a squaring transition. More generally, let T_i be the first step after T_{i-1} when the walk makes a squaring transition. Let $b_i = Y_{T_i-1} - Y_{T_{i-1}}$ be the contribution from the ith block (this is the part of the original walk covered by addition steps only). The block walk is then defined as the random process $Z_s = Y_{T_s} = 2^s Y_{T_0} + 2\sum_{i=1}^{s} 2^{s-i} b_i$. One can estimate the probability $B^s(u, v)$ of reaching a vertex v starting from a vertex u via a pseudo-random walk consisting of exactly s blocks (as opposed to a fixed number of steps). More precisely, assuming that $Z_0 = u$, $B^s(u, v)$ is the probability that $Z_s = v$. Obtaining good upper and lower bounds for these probabilities is possible for the following two reasons: if S is the set of blocks for which $b_i = 0$ or 1, i.e., the blocks of zero steps (two consecutive squarings), and the blocks of one multiplication-by-g step (a squaring followed by multiplication by g followed by another squaring) then one can separate the total contribution from these special blocks and conditions on the total contribution of the remaining blocks. Calculating this conditional probability amounts to calculating the probability of a given integer w being represented as $w = \sum_{i=1}^{s} 2^{s-i} b_i$ where $b_i = 0$ or 1. Using the uniqueness of the binary representation of w, one can determine uniquely the contribution of each block from the set S and thus, establish strong upper bounds on this conditional probability. This allows to deduce (via Plancherel's formula from Fourier analysis and the

fact that the contributions b_i of the blocks are independent considered as random variables) that $B^s(u, v)$ is close to uniform for s that is polylogarithmic in $\log|\mathbf{G}|$ which shows rapid mixing for the block walk. Here, the squaring step plays a key role.

It is difficult to generalize the block-walk method of [21, 22] in the additive Pollard rho setting since we have no natural choice for the separating move (as is the squaring step in the classical Pollard rho). Even if one declares one of the existing steps as separating, one still has no analogue of the uniqueness of the binary expansion. One can work harder and estimate the number of representations of w as a combination of the remaining (non-separating) steps, but eventually, we were unable to obtain an asymptotically useful bound. From that point of view, the problem of analyzing Pollard rho equipped with r-adding walks looks harder than analyzing the original Pollard rho using a mixed walk. Before stating the main result, we make two separate comments that will motivate the precise formulation chosen for the complexity analysis.

3.2 Randomization over the addition steps

Suppose that one is interested in analyzing the case of Pollard rho for additive walks. One then loses the important property that the squaring step contributes to the rapid mixing properties of the Pollard rho walk. When no squaring is used, there exist special cases in which it might take as many as $|\mathbf{G}|$ steps for the walk to even reach a certain element of the group.

To make this precise, note that each choice of the r-tuple $(s_1, \ldots, s_r)$ of transition elements gives rise to a stopping time random variable $T_{s_1,\ldots,s_r}$, namely, the first time when a collision is obtained in the walk. More precisely, let

$$T_{s_1,\ldots,s_r} = \min\{j > 0 : \exists i < j \text{ such that } X_i = X_j\},$$

where $(s_1, \ldots, s_r)$ are the transition steps in the Pollard rho walk.

Here, the distribution of $T_{s_1,\ldots,s_r}$ depends on the source of randomness for the Pollard rho walk. For Model 1, this source is the choice of a random partition of $\mathbf{G}$ into r disjoint sets whereas for Model 2, it is the random choice of a transition element at each step in the walk. In either case, we are interested in showing that $T_{s_1,\ldots,s_r}$ is bounded by some appropriate upper bound with either probability at least 50% (or any fixed probability, independent of $|\mathbf{G}|$) or with high probability (i.e., probability $1-\mathcal{O}(|\mathbf{G}|^{-c})$ for some $c > 0$) over the desired source of randomness. For instance, one might want to show that $T_{s_1,\ldots,s_r}$ is less than a constant times $\sqrt{|\mathbf{G}|}$ with probability more than 50% for an r-tuple $(s_1, \ldots, s_r)$ of transition steps. Yet, in the r-adding walk case, it is unreasonable to expect such a result to hold for a fixed r-tuple $(s_1, \ldots, s_r)$ as there might be very degenerate choices

that do not even allow one to reach every element of the group $\mathbf{G}$ in $\mathcal{O}(\sqrt{|\mathbf{G}|})$ steps. For instance, consider $\mathbf{G} = (\mathbf{Z}/N\mathbf{Z}, +)$ for some integer $N > 0$ and suppose that $r = \mathcal{O}(\log N)$. Consider the transition elements $s_i = i \bmod N$ for $i = 1, \ldots, r$. It is clear that if the walk starts at the zero element, it cannot reach the element $N - 1 \bmod N$ in time less than $N/\log N$. This degeneracy leads to a poor mixing time for the Pollard rho walk for this particular choice of steps. One way to remedy this issue is to establish the expected upper bound on the stopping time $T_{s_1,\ldots,s_r}$ with high probability over the random choice of the transition elements $(s_1, \ldots, s_r)$ as well as over the source of randomness of Model 2.

3.3 Dependency on *r*

One expects that by increasing the size of the set of precomputed points that can be added to the current point in the iteration function results in a pseudo-random walk behaving more like a truly random walk. This was experimentally shown to be true by Teske [36] and is made more precise by Bernstein who showed, using a heuristic argument [4, Appendix B] refining the analysis from [8], that the expected number of steps to reach a collision when using an r-adding walk is

$$\sqrt{\frac{\pi|\mathbf{G}|}{2(1-\sum_{j=1}^{r} p_j^2)}},$$

where typically $p_j \approx \frac{1}{r}$ (see also [5]). Hence, the use of an r-adding walk results in a bound that is larger than the birthday bound $\sqrt{\pi|\mathbf{G}|/2}$ by a factor of $\sqrt{r/(r-1)}$, so the larger r is, the closer the expected bound is to the birthday bound. This argument is extended in [7] when using mixed walks (walks that have both multiplication and squaring steps). The expected number of steps for reaching a collision is then

$$\sqrt{\frac{\pi|\mathbf{G}|}{2(1-p_D^2-\sum_{j=1}^{r} p_j^2)}},$$

where $p_D = 1 - \sum_{i=1}^{r} p_i$ is the probability of choosing a squaring step. For instance, the original mixed walk used by Pollard is expected to differ from the birthday bound by a factor of $\sqrt{3/2} \approx 1.22$.

A major question is then how r should depend on $|\mathbf{G}|$. Assuming that r does not depend on $|\mathbf{G}|$, we note that an argument of Greenhalgh [15] as extended by Hildebrand [20, Theorem 2] (see also Theorem 4.8) establishes lower bounds on the mixing time that are exponential (in $\log|\mathbf{G}|$) as opposed to mixing times that are polynomial (in $\log|\mathbf{G}|$) in the case when r depends logarithmically on $|\mathbf{G}|$

(see Theorem 4.4). One would then expect that in the case when the mixing time is poor, the number of steps to achieve a collision must be far from the birthday bound. We thus allow r to depend logarithmically on $|\mathbf{G}|$.

3.4 Main theorem

Similarly to the case of the classical Pollard rho algorithm with three transitions, proving anything under Model 1 seems hopeless. Our main theorem shows that such a result for r-adding walks can indeed be shown under Model 2 with an asymptotic upper bound $\sqrt{|\mathbf{G}|\log|\mathbf{G}|}$ on the number of steps (with probability greater than 50%).

Theorem 3.1. *Let $a > 1$, $\delta > 0$ and $\gamma > 0$ be real numbers. There exists $n_0 \geq 0$ with the following property: if $\mathbf{G}$ is a finite cyclic group of prime order $|\mathbf{G}| \geq n_0$ and $\kappa > 1$ is a real number then*

$$\Pr_{\substack{s_1,\ldots,s_r \\ \text{random walk}}}\left[T_{s_1,\ldots,s_r} \leq \sqrt{\frac{(2+\delta)\kappa}{e}|\mathbf{G}|\log|\mathbf{G}|}\right] \geq 1 - e^{-\kappa} - \frac{1}{|\mathbf{G}|^{\gamma}},$$

where $r = \lfloor(\log|\mathbf{G}|)^a\rfloor$. Here, the probability is taken over a uniformly random choice of an r-tuple $(s_1, \ldots, s_r)$ of distinct elements of $\mathbf{G}$ and over the randomness of Model 2.

The implications of this theorem are stated in the following corollary.

Corollary 3.2. *Let $a > 1$ be a real number. If $\mathbf{G}$ is a finite group that is cyclic of prime order and if $r = \lfloor(\log|\mathbf{G}|)^a\rfloor$ then solving the discrete logarithm problem with the Pollard rho method using an r-adding walk requires*

(i) $\mathcal{O}(\sqrt{|\mathbf{G}|\log|\mathbf{G}|})$ *steps with probability ≥ 0.5 as $|\mathbf{G}| \to \infty$,*

(ii) $\mathcal{O}(\sqrt{|\mathbf{G}|}\log|\mathbf{G}|)$ *steps with high probability, i.e., if $\gamma > 0$ is any fixed real number (independent of $|\mathbf{G}|$) then the probability of not finding a collision in $\mathcal{O}(\sqrt{|\mathbf{G}|}\log|\mathbf{G}|)$ steps is bounded by $\mathcal{O}(|\mathbf{G}|^{-\gamma})$ as $|\mathbf{G}| \to \infty$,*

where the probability is over the choice of uniformly random r-tuple $(s_1, \ldots, s_r)$ of distinct elements of $|\mathbf{G}|$ and the randomness in Model 2.

Proof. This follows immediately from Theorem 3.1 when we fix a constant $\gamma > 0$ and consider $\kappa = \log\frac{2|\mathbf{G}|^{\gamma}}{|\mathbf{G}|^{\gamma}-1}$ for the first part and $\kappa = \gamma\log|\mathbf{G}|$ for the second part. □

To prove the theorem, we first need mixing time estimates for random walks on the group $\mathbf{G}$. Such estimates over a random choice of r independent adding steps were established by Dou and Hildebrand [11, 19]. The idea then is the following: given the mixing time τ (i.e., the number of steps needed to make the end point of the walk look uniformly random), we make t_0 initial steps and then τ additional steps. Since τ is a mixing time, the probability of the end point of the walk being any of the t_0 initial points is $\frac{t_0}{|\mathbf{G}|}$, i.e., the probability of failing to produce a collision at this step should be bounded by $1 - \frac{t_0}{|\mathbf{G}|}$. If no collision has occurred, we perform another τ steps and calculate the probability of failure. We continue until the probability of failure becomes less than $e^{-\kappa}$. Suppose we have done s such iterations (performing τ steps s times after the original t_0 steps). We then need to minimize the total number of $t_0 + \tau s$ steps subject to the constraint that the failure probability is smaller than $e^{-\kappa}$. By solving this optimization problem, we see that it takes $\mathcal{O}(\sqrt{|\mathbf{G}|\log|\mathbf{G}|})$ steps to produce a collision with probability at least $1 - e^{-\kappa}$.

4 Random walks on groups and mixing times

Let $\mathbf{G}$ be a finite abelian group, let P be a probability distribution on $\mathbf{G}$ and let U be the uniform distribution on $\mathbf{G}$. Following [19], we define the statistical distance between the probability distribution P and the uniform distribution U as

$$\|P - U\| := \frac{1}{2}\sum_{s\in\mathbf{G}}\left|P(s) - \frac{1}{|\mathbf{G}|}\right|.$$

Given two real-valued functions $f\colon \mathbf{G} \to \mathbb{R}$ and $g\colon \mathbf{G} \to \mathbb{R}$, we define their convolution as

$$(f \star g)(x) := \sum_{y\in\mathbf{G}} f(xy^{-1})g(y).$$

As the convolution is associative, the m-fold convolution $f \star \cdots \star f$ is well-defined and we denote it by $f^{\star m}$. Let r be a positive integer (that may or may not depend on $|\mathbf{G}|$) and let $s_1, \ldots, s_r \in \mathbf{G}$ be a sequence of r distinct elements of $\mathbf{G}$ (we refer to these elements as the transition steps). Let $p_1, \ldots, p_r$ be a set of positive real numbers such that $\sum_{i=1}^{r} p_i = 1$ (we refer to these numbers as transition probabilities). Let $P_{s_1,\ldots,s_r}$ be the distribution defined by

$$P_{s_1,\ldots,s_r}(s) := \begin{cases} p_i & \text{if } s = s_i \text{ for some } i = 1, \ldots, r, \\ 0 & \text{otherwise.} \end{cases}$$

In the case of the r-adding walk version of Pollard rho under Model 2, we take $p_i = 1/r$ for every $i = 1, \ldots, r$. Note that $P^{\star m}_{s_1,\ldots,s_r}$ is the distribution of the end

point of an m-step random walk starting from the identity element of the group $\mathbf{G}$ (this follows, e.g., by induction on m).

Definition 4.1. Given $\epsilon > 0$ and a sequence $s_1, \ldots, s_r$ of transition steps, we define the ϵ-mixing time $\tau_{s_1,\ldots,s_r}(\epsilon)$ with respect to that sequence to be the smallest integer m such that $\| P^{\star m}_{s_1,\ldots,s_r} - U \| < \epsilon$.

Remark 4.2. Using the fact that $s_1, \ldots, s_r$ generate $\mathbf{G}$ (since $\mathbf{G}$ is of prime order), using spectral analysis of the adjacency matrices of the Cayley graph constructed from $\{s_1, \ldots, s_r\}$, one can show that the mixing time $\tau_{s_1,\ldots,s_r}(\epsilon)$ is well-defined.

Remark 4.3. We cannot find a reasonable bound for $\tau_{s_1,\ldots,s_r}(\epsilon)$ for every r-tuple $(s_1, \ldots, s_r)$ of transition steps. Yet, for a uniformly random r-tuple among all $\frac{|\mathbf{G}|!}{(|\mathbf{G}|-r)!}$ r-tuples of distinct elements of $\mathbf{G}$, one could expect a reasonable upper bound. This can be formalized using Markov's inequality as well as bounds on the expectation of the statistical difference between the distribution of the end-point of the mth step of the random walk and the uniform distribution (due to Hildebrand). This is indeed the approach that we take.

The next theorem shows that if we allow polylogarithmic dependence of r on $|\mathbf{G}|$ then one does indeed get a polynomial (in $\log|\mathbf{G}|$) mixing time.

Theorem 4.4 ([11, Theorem 1]). *Let $r = \lfloor (\log|\mathbf{G}|)^a \rfloor$ for some constant $a > 1$ and let $\epsilon' > 0$ be given. Suppose that $m > \frac{a}{a-1} \frac{\log|\mathbf{G}|}{\log r}(1+\epsilon')$. Then*

$$\mathop{\mathbb{E}}_{s_1,\ldots,s_r} \left[\| P^{*m}_{s_1,\ldots,s_r} - U \| \right] \to 0 \quad \text{as } |\mathbf{G}| \to \infty,$$

where the probability is taken over a uniformly random r-tuple $(s_1, \ldots, s_r)$ of distinct elements of $\mathbf{G}$.

Remark 4.5. For our particular application, the version stated above is not sufficient as it does not quantify the rate of convergence of the expectation as $|\mathbf{G}| \to \infty$. Yet, we should point out that such a quantification is implicit in the proof by Dou and Hildebrand. We state and prove an effective version in the next section and apply this version to obtain upper bounds on the mixing time $\tau_{s_1,\ldots,s_r}(\epsilon)$ that holds with high probability over the choice of the r-tuple $(s_1, \ldots, s_r)$.

Remark 4.6. We note that Hildebrand's bound is optimal in the following sense: it is shown in [19, Theorem 3] that if $r = \lfloor (\log|\mathbf{G}|)^a \rfloor$ for some constant $a < 1$ then for any fixed positive real number b, the distance $\| P^{*m}_{s_1,\ldots,s_r} - U \| \to 1$ as $|\mathbf{G}| \to \infty$ for $m = \lfloor (\log|\mathbf{G}|)^b \rfloor$ and any choice $s_1, \ldots, s_r \in \mathbf{G}$ of transition steps.

Finally, we note that if r is independent of $|\mathbf{G}|$ then the mixing time becomes exponential as shown by the following two theorems establishing upper and lower bounds, respectively.

Theorem 4.7 ([19, Theorem 1]). *Suppose that $r \geq 2$ is a fixed positive integer and let $p_1, \ldots, p_r$ be fixed transition probabilities as above. Given $\epsilon > 0$, there exists a constant γ that depends on r, ϵ and the p_i's, but not on $|\mathbf{G}|$ such that*

$$\mathop{\mathbb{E}}_{s_1,\ldots,s_r}\left[\| P^{*m}_{s_1,\ldots,s_r} - U \|\right] < \epsilon$$

for $m = \lfloor \gamma |\mathbf{G}|^{2/(r-1)} \rfloor$ where the expectation is taken over a uniformly random r-tuple $(s_1, \ldots, s_r)$ of distinct elements of $\mathbf{G}$.

Theorem 4.8 ([20, Theorem 2]). *Let r be constant (independent of $|\mathbf{G}|$) and let $p_i = \frac{1}{r}$ for $i = 1, \ldots, r$. Let $\delta < \frac{1}{2}$ be fixed. There exists a value $\gamma > 0$ (independent of $|\mathbf{G}|$) such that if $m < \gamma |\mathbf{G}|^{2/(r-1)}$ then $\| P^{*m}_{s_1,\ldots,s_r} - U \| > \delta$ for any r-tuple $(s_1, \ldots, s_r)$ of distinct elements of $\mathbf{G}$.*

4.1 Upper bounds on mixing times

The mixing time $\tau_{s_1,\ldots,s_r}(\epsilon)$ can get as large as $|\mathbf{G}|$ for certain degenerate r-tuples $(s_1, \ldots, s_r)$ of transition steps. We thus need a way to show that with high probability over a randomly chosen r-tuple $(s_1, \ldots, s_r)$ of distinct elements of $\mathbf{G}$, the mixing time can be bounded by a sufficiently strong upper bound. The following definition is helpful in order to make this rigorous:

Definition 4.9. Let $\epsilon > 0$ be a real number and let m be a positive integer. We say that an r-tuple $(s_1, \ldots, s_r)$ of distinct elements of $\mathbf{G}$ is (ϵ, m)-*faulty* if $\| P^{\star m}_{s_1,\ldots,s_r} - U \| > \epsilon$ or equivalently, if $\tau_{s_1,\ldots,s_r}(\epsilon) > m$.

Using the work of Dou and Hildebrand [11] and Markov's inequality, one can prove the following:

Lemma 4.10. *Let $a > 1$ be any real number and let ϵ' be a real number that satisfies $0 < \epsilon' < \frac{(a-1)}{e} \log\log|\mathbf{G}| - 1$. Let*

$$r = \lfloor (\log|\mathbf{G}|)^a \rfloor \quad \text{and} \quad m = \left\lceil \frac{\log|\mathbf{G}|}{(a-1)\log\log|\mathbf{G}|}(1+\epsilon') \right\rceil.$$

There exists $n_0' > 0$ such that if $|\mathbf{G}| \geq n_0'$ then for any $\epsilon > 0$, we have

$$\Pr_{s_1,\ldots,s_r}\left[(s_1, \ldots, s_r) \text{ is } (\epsilon, m)\text{-faulty}\right] = \Pr_{s_1,\ldots,s_r}\left[\| P^{\star m}_{s_1,\ldots,s_r} - U \| > \epsilon\right] < \frac{3}{4\epsilon^2 |\mathbf{G}|^{\epsilon'}},$$

where the probability is taken over a uniformly random r-tuple $(s_1, \ldots, s_r)$ of distinct elements of $\mathbf{G}$.

Proof. We follow the proof of [11, Theorem 1]. Throughout, we omit the explicit reference to the floor and ceiling notation that does not affect any of the conclusions. It is shown in [11, p. 996] that

$$\mathop{\mathbb{E}}_{s_1,\ldots,s_r}\left[\| P^{\star m}_{s_1,\ldots,s_r} - U \|^2\right] \leq \frac{3}{4}\frac{|\mathbf{G}|(em)^m}{r^m}, \tag{4.1}$$

whenever $|\mathbf{G}|$ is sufficiently large, i.e., whenever $|\mathbf{G}| \geq n_0'$ for some $n_0' > 0$. Letting

$$d = \frac{a}{a-1}\frac{1+\epsilon'}{\log r} = \frac{1+\epsilon'}{(a-1)\log\log|\mathbf{G}|},$$

note that $(em)^m = |\mathbf{G}|^{d(\log d+1)}$ and $r^m = e^{m(a-1)\log\log|\mathbf{G}|}$. The right side of (4.1) becomes

$$\frac{3|\mathbf{G}|(em)^m}{4r^m} = \frac{3e^{\log|\mathbf{G}|}|\mathbf{G}|^{d(\log d+1)}}{4e^{m(a-1)\log\log|\mathbf{G}|}} = \frac{3}{4}|\mathbf{G}|^{-\epsilon'+d(\log d+1)}.$$

If $\epsilon' < \frac{(a-1)\log\log|\mathbf{G}|}{e} - 1$ then $\log d < -1$, hence, $d(\log d + 1) < 0$ and (4.1) then implies

$$\mathop{\mathbb{E}}_{s_1,\ldots,s_r}\left[\| P^{\star m}_{s_1,\ldots,s_r} - U \|^2\right] \leq \frac{3}{4}\frac{1}{|\mathbf{G}|^{\epsilon'-d(\log d+1)}} < \frac{3}{4|\mathbf{G}|^{\epsilon'}}. \tag{4.2}$$

Using Markov's inequality, we obtain

$$\Pr_{s_1,\ldots,s_r}\left[\| P^{\star m}_{s_1,\ldots,s_r} - U \|^2 > \epsilon^2\right] < \frac{3}{4\epsilon^2|\mathbf{G}|^{\epsilon'}}.$$

Since the statistical distance is non-negative, the above inequality is equivalent to

$$\Pr_{s_1,\ldots,s_r}\left[\| P^{\star m}_{s_1,\ldots,s_r} - U \| > \epsilon\right] < \frac{3}{4\epsilon^2|\mathbf{G}|^{\epsilon'}}. \qquad \square$$

5 Application to Pollard rho – Proof of Theorem 3.1

We prove Theorem 3.1 in two different steps: (1) we establish collision bounds in terms of mixing times; (2) we combine the previous bounds via a simple result from probability theory and carefully optimize for the parameters involved.

5.1 Collision bounds and mixing times

Our proof is based on the following argument: let $\epsilon > 0$ and the r-tuple of transition steps $(s_1, \ldots, s_r)$ be fixed. For notational convenience, we substitute $\tau = \tau_{s_1,\ldots,s_r}(\epsilon)$. Given $t \geq 0$, we consider the probability $p(\epsilon, t, t_0)$ that no collision has occurred after the first $t_0 + (t+1)\tau$ steps of the walk. For instance, in case $t = 0$, we make $t_0 + \tau$ steps of the walk and compare the resulting end-point with the first t_0 elements of the walk. We can look at the probability $p(\epsilon, 0, t_0)$ and by definition of the mixing time τ,

$$p(\epsilon, 0, t_0) \leq 1 - \frac{t_0(1 - 2\epsilon|\mathbf{G}|)}{|\mathbf{G}|}.$$

More generally, the probability that the $(t_0 + (t+1)\tau)$th element of the walk does not collide with any of the first $t_0 + t\tau$ elements is at most

$$1 - \frac{(t_0 + t\tau)(1 - 2\epsilon|\mathbf{G}|)}{|\mathbf{G}|}.$$

Using $1 - x \leq e^{-x}$ for $x < 1$, we obtain

$$\begin{aligned} p(\epsilon, t, t_0) &\leq \Big(1 - \frac{(t_0 + t\tau)(1 - 2\epsilon|\mathbf{G}|)}{|\mathbf{G}|}\Big) p(\epsilon, t-1, t_0) \\ &\leq \exp\Big(-\frac{(t_0 + t\tau)(1 - 2\epsilon|\mathbf{G}|)}{|\mathbf{G}|}\Big) p(\epsilon, t-1, t_0). \end{aligned}$$

By induction on t, we prove the following:

Lemma 5.1. *Let $\epsilon > 0$ be a real number and let $(s_1, \ldots, s_r)$ be a fixed r-tuple of distinct elements of $\mathbf{G}$. Let $\tau = \tau_{s_1,\ldots,s_r}(\epsilon)$ be the mixing time introduced in Definition* 4.1. *For any positive integers $t \geq 0$ and t_0 we have*

$$p(\epsilon, t, t_0) \leq \exp\Big(-\frac{(1 - 2\epsilon|\mathbf{G}|)(t+1)(2t_0 + t\tau)}{2|\mathbf{G}|}\Big) =: B(\epsilon, t, t_0).$$

The above lemma bounds the probability that a collision is obtained after $t_0 + (t+1)\tau$ steps. The next step is to optimize the integer parameters t_0 and t. Thus, the probability of *obtaining a collision* after at most $t(\epsilon, t, t_0) := t_0 + (t+1)\tau(\epsilon)$ Pollard rho steps is lower bounded by

$$\Pr_{\text{random walk}}\big[T_{s_1,\ldots,s_r} \leq t(\epsilon, t, t_0)\big] \geq 1 - B(\epsilon, t, t_0).$$

For any $\epsilon < \frac{1}{2|\mathbf{G}|}$ and any t_0 (keeping in mind that $\kappa > 1$ by hypothesis), we first determine the minimal value of t for which the probability of failure (not obtaining a collision after $t(\epsilon, t, t_0)$ steps) is at most $e^{-\kappa}$. In other words,

$$B(\epsilon, t, t_0) = \exp\Big(-\frac{(1-2\epsilon|\mathbf{G}|)(t+1)(2t_0+t\tau)}{2|\mathbf{G}|}\Big) \leq e^{-\kappa}$$
$$\iff \frac{\tau}{2}t^2 + \Big(\frac{\tau}{2} + t_0\Big)t + t_0 - \frac{\kappa|\mathbf{G}|}{1-2\epsilon|\mathbf{G}|} \geq 0.$$

Here, we have used the hypothesis $\epsilon < \frac{1}{2|\mathbf{G}|}$ which implies that the discriminant of the above quadratic polynomial in t is positive, hence, by solving the quadratic inequality, we obtain

$$t \geq \frac{-(\frac{\tau}{2} + t_0) + \sqrt{\frac{2\kappa|\mathbf{G}|}{1-2\epsilon|\mathbf{G}|}\tau + (\frac{\tau}{2} - t_0)^2}}{\tau} =: t'.$$

This means that $t_{\min} = \lceil t' \rceil$ is the minimal possible value for t that yields a probability of failure smaller than $e^{-\kappa}$ given ϵ, $(s_1, \ldots, s_r)$ and t_0. Hence, the number of Pollard rho steps necessary for producing a collision can be bounded by

$$t(\epsilon, t_{\min}, t_0) \leq \sqrt{\frac{2\kappa|\mathbf{G}|}{1-2\epsilon|\mathbf{G}|}\tau + \Big(\frac{\tau}{2} - t_0\Big)^2} + \frac{\tau}{2}.$$

The value of t_0 that minimizes the above bound is $t_0 = \lfloor \frac{\tau}{2} \rfloor$. Hence,

$$\min_{t,t_0} t(\epsilon, t, t_0) \leq \sqrt{\frac{2\kappa|\mathbf{G}|}{1-2\epsilon|\mathbf{G}|}\tau + \frac{1}{4}} + \frac{\tau}{2} =: \mathrm{t}_{s_1,\ldots,s_r}(\epsilon).$$

Here, we intentionally write $s_1, \ldots, s_r$ in the subscript to remind the reader of the dependency on the transition steps. Finally,

$$\Pr_{\text{random walk}}\big[T_{s_1,\ldots,s_r} \leq \mathrm{t}_{s_1,\ldots,s_r}(\epsilon)\big] \geq 1 - e^{-\kappa}. \tag{5.1}$$

5.2 Completing the proof

As we proceed with the proof, we will show what inequalities the n_0 needs to satisfy. First, suppose that n_0 satisfies

$$\frac{a-1}{e}\log\log n_0 - 1 > 0,$$

as well as $n_0 \geq n_0'$ where n_0' is the bound from Lemma 4.10 and suppose that $|\mathbf{G}| \geq n_0$. Let ϵ' be any real number that satisfies

$$0 < \epsilon' < \frac{a-1}{e} \log\log n_0 - 1$$

(the existence of such ϵ' is guaranteed by the above inequality) and let

$$m = \left\lfloor \frac{\log|\mathbf{G}|}{(a-1)\log\log|\mathbf{G}|}(1+\epsilon') \right\rfloor.$$

Let $\epsilon > 0$ be a real number that satisfies $\epsilon < \frac{1}{2|\mathbf{G}|}$. Lemma 4.10 yields a bound on the probability of drawing an (ϵ, m)-faulty choice of $(s_1, \ldots, s_r)$ out of all r-tuples of distinct elements in $\mathbf{G}$. To complete the proof of Theorem 3.1, we need to combine (5.1) and (4.2) to get the desired result. We do this via standard union bounds from probability theory by using that if A, B_1 and B_2 are three events such that $A \Rightarrow \neg B_1 \vee \neg B_2$ then

$$\Pr[A] \leq \Pr[\neg B_1] + \Pr[\neg B_2]. \tag{5.2}$$

We would like to apply (5.2) to A being the event

$$\text{Event } A: \quad T_{s_1,\ldots,s_r} > c\sqrt{\kappa|\mathbf{G}|\log|\mathbf{G}|}$$

for some constant $c > 0$ that we specify below. Moreover, let B_1 be the event

$$\text{Event } B_1: \quad T_{s_1,\ldots,s_r} \leq \mathrm{t}_{s_1,\ldots,s_r}(\epsilon),$$

and let B_2 be the event

$$\text{Event } B_2: \quad \tau_{s_1,\ldots,s_r}(\epsilon) \leq m,$$

where m is the value from Lemma 4.10. In order to apply (5.2), we only need to choose the parameters ϵ and ϵ' in such a way that $A \Rightarrow \neg B_1 \vee \neg B_2$.

Assuming that the events B_1 and B_2 both hold, we obtain

$$T_{s_1,\ldots,s_r} \leq \sqrt{\frac{2\kappa|\mathbf{G}|}{1-2\epsilon|\mathbf{G}|}m + \frac{1}{4}} + \frac{m}{2}.$$

Since $\epsilon' < \frac{(a-1)}{e}\log\log|\mathbf{G}| - 1$, we have $m < \frac{\log|\mathbf{G}|}{e} + 1$. Substituting this in the above inequality, we obtain

$$\begin{aligned} T_{s_1,\ldots,s_r} &\leq \sqrt{\frac{2\kappa|\mathbf{G}|}{e(1-2\epsilon|\mathbf{G}|)}(\log|\mathbf{G}|+e) + \frac{1}{4}} + \frac{\log|\mathbf{G}|+e}{2e} \\ &\overset{(*)}{\leq} \sqrt{\frac{2\kappa|\mathbf{G}|}{e(1-2\epsilon|\mathbf{G}|)}(\log|\mathbf{G}|+3)}. \end{aligned}$$

Here, the inequality $(*)$ holds since we add an extra condition on n_0 that is deduced below. First let us notice that $\sqrt{a}+b=\sqrt{a+b^2+2b\sqrt{a}}$, for all positive real numbers a, b. Applying this for

$$a=\frac{2\kappa|\mathbf{G}|}{e(1-2\epsilon|\mathbf{G}|)}(\log|\mathbf{G}|+e)+\frac{1}{4} \quad \text{and} \quad b=\frac{\log|\mathbf{G}|+e}{2e},$$

we observe that in order for $(*)$ to hold, we need to prove that

$$\frac{1}{4}+\Big(\frac{\log|\mathbf{G}|+e}{2e}\Big)^2+\frac{\log|\mathbf{G}|+e}{e}\sqrt{\frac{2\kappa|\mathbf{G}|}{e(1-2\epsilon|\mathbf{G}|)}(\log|\mathbf{G}|+e)+\frac{1}{4}} < \frac{2\kappa|\mathbf{G}|(3-e)}{e(1-2\epsilon|\mathbf{G}|)}.$$

Equivalently, we need to show that

$$\frac{e(1-2\epsilon|\mathbf{G}|)}{8\kappa|\mathbf{G}|}+\frac{(\log|\mathbf{G}|+e)^2(1-2\epsilon|\mathbf{G}|)}{8e\kappa|\mathbf{G}|} + \frac{(\log|\mathbf{G}|+e)^{3/2}\sqrt{e(1-2\epsilon|\mathbf{G}|)}}{e\sqrt{2\kappa|\mathbf{G}|}}\sqrt{1+\frac{e(1-2\epsilon|\mathbf{G}|)}{8\kappa|\mathbf{G}|(\log|\mathbf{G}|+e)}} < 3-e$$

and as $1-2\epsilon|\mathbf{G}|<1$ and $\kappa>\log 2$, it suffices to show that one can choose n_0 such that for any $\mathbf{G}$ with $|\mathbf{G}|\geq n_0$,

$$\frac{e}{8|\mathbf{G}|\log 2}+\frac{(\log|\mathbf{G}|+e)^2}{8e\log 2\cdot|\mathbf{G}|} + \frac{(\log|\mathbf{G}|+e)^{3/2}}{\sqrt{2e\log 2\cdot|\mathbf{G}|}}\sqrt{1+\frac{e}{8|\mathbf{G}|\log 2(\log|\mathbf{G}|+e)}} < 3-e.$$

Now, if our n_0 is chosen such that each term

$$\frac{e}{8n_0\log 2}<\frac{3-e}{3}, \qquad \frac{(\log n_0+e)^2}{8e\log 2\cdot n_0}<\frac{3-e}{3}$$
$$\frac{(\log n_0+e)^{3/2}}{\sqrt{2e\log 2\cdot n_0}}\sqrt{1+\frac{e}{8n_0\log 2(\log n_0+e)}}<\frac{3-e}{3},$$

then the above inequality holds for any $\mathbf{G}$ with $|\mathbf{G}|\geq n_0$.

The largest term is upper bounded by $(\log n_0+e)^{3/2}/\sqrt{n_0}$. If we impose the extra condition that this bound is also less than $(3-e)/3$ (i.e., the extra condition that n_0 should satisfy $9(\log n_0+e)^3<(3-e)^2 n_0$), all of the above inequalities will hold whenever $|\mathbf{G}|\geq n_0$.

We now specify the constant c: it should be chosen such that the above upper bound can be further bounded by $c\sqrt{\kappa|\mathbf{G}|\log|\mathbf{G}|}$. It is clear that $c = 2/e$ is too small. Yet, we observe that for any positive real number $\delta > 0$ (independent of $|\mathbf{G}|$), one can use $c = \sqrt{(2+\delta)/e}$ and as long as $\epsilon < (1 - \frac{2}{2+\delta})\frac{1}{2|\mathbf{G}|}$, i.e.,

$$T_{s_1,\ldots,s_r} \leq \sqrt{\frac{2+\delta}{e}\kappa|\mathbf{G}|\log|\mathbf{G}|}.$$

Next, for the specified $\gamma > 0$, we would like to choose ϵ such that the upper bound on $\Pr(\neg B_2)$ established in Lemma 4.10 is upper bounded by $|\mathbf{G}|^{-\gamma}$), e.g., that

$$\frac{3}{4\epsilon^2|\mathbf{G}|^{\epsilon'}} < \frac{1}{|\mathbf{G}|^{\gamma}}$$

which is achieved as long as $\epsilon > \sqrt{3/4}\,|\mathbf{G}|^{(\gamma-\epsilon')/2}$.

To summarize, if we want to choose ϵ so that we guarantee simultaneously the following two conditions:

(i) $B_1 \wedge B_2 \Rightarrow \neg A$,

(ii) $\Pr(\neg B_2) < |\mathbf{G}|^{-\gamma}$,

then we need the lower bound on ϵ (namely, $\sqrt{3/4}\,|\mathbf{G}|^{(\gamma-\epsilon')/2}$) to not exceed the upper bound (namely, $(1 - \frac{2}{2+\delta})\frac{1}{2|\mathbf{G}|}$). This is achieved as long as $\epsilon' > \gamma + 2$. Since the only constraint on ϵ' is $\epsilon' < (a-1)/e \log\log|\mathbf{G}| - 1$, if we choose n_0 sufficiently large so that

$$\frac{(a-1)}{e}\log\log n_0 - 1 > \gamma + 2, \tag{5.3}$$

$$9\,(\log n_0 + e)^3 < (3-e)^2 n_0, \tag{5.4}$$

$$n_0 \geq n_0' \tag{5.5}$$

(where n_0' is the bound from Lemma 4.10), then for any $\mathbf{G}$ for which $|\mathbf{G}| \geq n_0$, the two conditions (i) and (ii) will hold. Thus, the only conditions that we need for n_0 are the inequalities (5.3), (5.4) and (5.5).

Finally, if the events B_1 and B_2 both occur then

$$T_{s_1,\ldots,s_r} < \sqrt{\frac{2+\delta}{e}\kappa|\mathbf{G}|\log|\mathbf{G}|},$$

so the event A, i.e., $T_{s_1,\ldots,s_r} > \sqrt{\frac{2+\delta}{e}\kappa|\mathbf{G}|\log|\mathbf{G}|}$, is impossible. The union bound then implies

$$\Pr_{\substack{s_1,\ldots,s_r \\ \text{random walk}}}\left[T_{s_1,\ldots,s_r} > \sqrt{\frac{2+\delta}{e}\kappa|\mathbf{G}|\log|\mathbf{G}|}\right] < e^{-\kappa} + \frac{1}{|\mathbf{G}|^{\gamma}},$$

i.e.,

$$\Pr_{\substack{s_1,\ldots,s_r \\ \text{random walk}}}\left[T_{s_1,\ldots,s_r} \leq \sqrt{\frac{2+\delta}{e}\kappa|\mathbf{G}|\log|\mathbf{G}|}\right] > 1 - e^{-\kappa} - \frac{1}{|\mathbf{G}|^{\gamma}},$$

which concludes the proof of Theorem 3.1.

6 Conclusions

In the past few years, there has been a lot of attempts dedicated to rigorously proving the asymptotic run-time of generic algorithms to solve the discrete logarithm problem based on the Pollard's rho method [21–23, 26, 29]. We rigorously prove collision bounds for general cyclic groups $\mathbf{G}$ of prime order for the most common variation of Pollard rho (currently used to solve the discrete logarithm problem on a generic elliptic curve), namely the Pollard rho method using additive walks. Using mixing time estimates from Dou and Hildebrand [11, 19], we are able to prove a collision bound of $\mathcal{O}(\sqrt{|\mathbf{G}|\log|\mathbf{G}|})$ with probability greater than 50%. We hope that, just as in the case of the original Pollard rho setting, this is only the first step in rigorously proving the correct asymptotic bound for the additive Pollard rho algorithm.

Acknowledgments. We are grateful to Arjen Lenstra for the careful reading of the draft and for the numerous helpful comments and discussions. We also thank Emmanuel Kowalski, Philippe Michel, Martijn Stam and Ramarathnam Venkatesan for various discussions. Finally, we thank the anonymous reviewer for improving the quality of the paper.

Bibliography

[1] L. M. Adleman, A subexponential algorithm for the discrete logarithm problem with applications to cryptography (abstract), in: *Foundations of Computer Science* (FOCS), IEEE Computer Society (1979), 55–60.

[2] L. M. Adleman, J. DeMarrais and M.-D. A. Huang, A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields, in: *Algorithmic Number Theory* (ANTS I), Lecture Notes in Comput. Sci. 877, Springer, Heidelberg (1994), 28–40.

[3] L. M. Adleman, J. DeMarrais and M.-D. A. Huang, A subexponential algorithm for discrete logarithms over hyperelliptic curves of large genus over GF(q), *Theoret. Comput. Sci.* **226** (1999), 7–18.

[4] D. V. Bailey et al., *Breaking ECC2K-130*, preprint (2009), `http://eprint.iacr.org/2009/541`.

[5] D. J. Bernstein and T. Lange, Two grumpy giants and a baby, in: *Algorithmic Number Theory* (ANTS-X), Mathematical Science Publishers, Berkeley (2013), 87–111.

[6] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra and P. L. Montgomery, Solving a 112-bit prime elliptic curve discrete logarithm problem on game consoles using sloppy reduction, *Int. J. Appl. Cryptogr.* **2** (2012), 212–228.

[7] J. W. Bos, T. Kleinjung and A. K. Lenstra, On the use of the negation map in the Pollard rho method, in: *Algorithmic Number Theory* (ANTS-IX), Lecture Notes in Comput. Sci. 6197, Springer, Heidelberg (2010), 67–83.

[8] R. P. Brent and J. M. Pollard, Factorization of the eighth Fermat number, *Math. Comp.* **36** (1981), 627–630.

[9] Certicom, Certicom announces elliptic curve cryptosystem (ECC) challenge winner, press release, 2002, `www.certicom.com/index.php/2002-press-releases/38-2002-press-releases/340-notre-dame-mathematician-solves-eccp-109-encryption-key-problem-issued-in-1997`.

[10] C. Diem, On the discrete logarithm problem in class groups of curves, *Math. Comp.* **80** (2011), 443–475.

[11] C. Dou and M. Hildebrand, Enumeration and random random walks on finite groups, *Ann. Probab.* **24** (1996), 987–1000.

[12] A. Enge, Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time, *Math. Comp.* **71** (2002), 729–742.

[13] P. Flajolet and A. M. Odlyzko, Random mapping statistics, in: *Eurocrypt 1989*, Lecture Notes in Comput. Sci. 434, Springer, Heidelberg (1990), 329–354.

[14] P. Gaudry, An algorithm for solving the discrete log problem on hyperelliptic curves, in: *Eurocrypt 2000*, Lecture Notes in Comput. Sci. 1807, Springer, Heidelberg (2000), 19–34.

[15] A. S. Greenhalgh, *Random walks on groups with subgroup invariance properties*, Ph.D. thesis, Stanford University, 1989.

[16] R. Harley, Elliptic curve discrete logarithms project, preprint (2000), `http://pauillac.inria.fr/~harley/ecdl/`.

[17] B. Harris, Probability distributions related to random mappings, *Ann. Math. Stat.* **31** (1960), 1045–1062.

[18] F. Hess, Computing relations in divisor class groups of algebraic curves over finite fields, preprint (2004), `www.staff.uni-oldenburg.de/florian.hess/publications/dlog.pdf`

[19] M. Hildebrand, Random walks supported on random points of Z/nZ, *Probab. Theory Related Fields* **100** (1994), 191–203.

[20] M. Hildebrand, A survey of results on random random walks on finite groups, *Probab. Surv.* **2** (2005), 33–63.

[21] J. H. Kim, R. Montenegro, Y. Peres and P. Tetali, A birthday paradox for Markov chains, with an optimal bound for collision in the Pollard rho algorithm for discrete logarithm, in: *Algorithmic Number Theory* (ANTS-VIII), Lecture Notes in Comput. Sci. 5011, Springer, Heidelberg (2008), 402–415.

[22] J. H. Kim, R. Montenegro, Y. Peres and P. Tetali, A birthday paradox for Markov chains, with an optimal bound for collision in the Pollard rho algorithm for discrete logarithm, *Ann. Appl. Probab.* **20** (2010), 495–521.

[23] J. H. Kim, R. Montenegro and P. Tetali, Near optimal bounds for collision in Pollard rho for discrete log, in: *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science* (FOCS '07), IEEE Computer Society (2007), 215–223.

[24] D. E. Knuth, *Seminumerical Algorithms*, 3rd ed., The Art of Computer Programming, Addison-Wesley, Reading, 1997.

[25] N. Koblitz, Elliptic curve cryptosystems, *Math. Comp.* **48** (1987), 203–209.

[26] S. D. Miller and R. Venkatesan, Spectral analysis of Pollard rho collisions, in: *Algorithmic Number Theory* (ANTS-VII), Lecture Notes in Comput. Sci. 4076, Springer, Heidelberg (2006), 573–581.

[27] S. D. Miller and R. Venkatesan, Non-degeneracy of Pollard rho collisions, *Int. Math. Res. Not.* **1** (2009), 1–10.

[28] V. S. Miller, Use of elliptic curves in cryptography, in: *Crypto 1985*, Lecture Notes in Comput. Sci. 218, Springer, Heidelberg (1986), 417–426.

[29] R. Montenegro and P. Tetali, How long does it take to catch a wild kangaroo?, in: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing* (STOC '09), ACM (2009), 553–560.

[30] P. L. Montgomery, Speeding the Pollard and elliptic curve methods of factorization, *Math. Comp.* **48** (1987), 243–264.

[31] National Security Agency, Fact sheet NSA Suite B Cryptography, 2009, `www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml`.

[32] J. M. Pollard, A Monte Carlo method for factorization, *BIT* **15** (1975), 331–334.

[33] J. M. Pollard, Monte Carlo methods for index computation (mod p), *Math. Comp.* **32** (1978), 918–924.

[34] J. M. Pollard, Kangaroos, monopoly and discrete logarithms, *J. Cryptology* **13** (2000), 437–447.

[35] V. Shoup, Lower bounds for discrete logarithms and related problems, in: *Eurocrypt 1997*, Lecture Notes in Comput. Sci. 1233, Springer, Heidelberg (1997), 256–266.

[36] E. Teske, On random walks for Pollard's rho method, *Math. Comp.* **70** (2001), 809–825.

[37] P. C. van Oorschot and M. J. Wiener, Parallel collision search with cryptanalytic applications, *J. Cryptology* **12** (1999), 1–28.

Received December 14, 2012; revised August 2, 2013; accepted December 23, 2013.

Author information

Joppe W. Bos, Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA.
E-mail: jbos@microsoft.com

Alina Dudeanu, Laboratory for Cryptologic Algorithms, EPFL, Lausanne, Switzerland.
E-mail: alina.dudeanu@epfl.ch

Dimitar Jetchev, Laboratory for Cryptologic Algorithms, EPFL, Lausanne, Switzerland.
E-mail: dimitar.jetchev@epfl.ch