Using Equivalence Classes to Accelerate Solving the Discrete Logarithm Problem in a Short Interval

Steven D. Galbraith* and Raminder S. Ruprai

[1] Mathematics Department, Auckland University, Auckland, New Zealand
s.galbraith@math.auckland.ac.nz
[2] Mathematics Department, Royal Holloway University of London, Egham, Surrey
TW20 0EX, UK .
raminder@email.com

Abstract. The Pollard kangaroo method solves the discrete logarithm problem (DLP) in an interval of size N with heuristic average case expected running time approximately $2\sqrt{N}$ group operations. It is well-known that the Pollard rho method can be sped-up by using equivalence classes (such as orbits of points under an efficiently computed group homomorphism), but such ideas have not been used for the DLP in an interval. Indeed, it seems impossible to implement the standard kangaroo method with equivalence classes.

The main result of the paper is to give an algorithm, building on work of Gaudry and Schost, to solve the DLP in an interval of size N with heuristic average case expected running time of close to $1.36\sqrt{N}$ group operations for groups with fast inversion. In practice the algorithm is not quite this fast, due to the usual problems with pseudorandom walks such as fruitless cycles. In addition, we present experimental results.

Keywords: discrete logarithm problem (DLP), elliptic curves, negation map, efficiently computable group homomorphisms.

1 Introduction

The discrete logarithm problem (DLP) in an interval is the problem: Given g, h in a group G and $N \in \mathbb{Z}_{>0}$ such that $h = g^n$ for some $0 \leq n \leq N$ (where N is less than the order of g), to compute n. This problem arises naturally in a number of contexts, for example the DLP with c-bit exponents (c-DLSE) [15, 23, 21], decryption in the Boneh-Goh-Nissim homomorphic encryption scheme [1], counting points on curves or abelian varieties over finite fields [14], the analysis of the strong Diffie-Hellman problem [3, 17], and side-channel or small subgroup attacks [16, 18].

One can solve the DLP in an interval using the baby-step-giant-step algorithm in at worst $2\sqrt{N}$ group operations (or, with minor modifications, with

* This work supported by EPSRC grant EP/D069904/1.

average case running time of $\sqrt{2N}$ group operations). But this method also requires $O(\sqrt{N})$ group elements of storage.

Pollard [24] developed the kangaroo algorithm precisely with this application in mind. Using distinguished points, van Oorschot and Wiener [22] (also see Pollard [26]) achieve a heuristic average case expected complexity of essentially $2\sqrt{N}$ group operations and low storage. We summarise this algorithm in Appendix A. Note that this algorithm has success probability of 1, as do all algorithms in this paper. These algorithms can also be parallelised (or distributed) with a linear speedup. For comparison, the Pollard rho method [24] has heuristic expected running time of $\sqrt{\pi r/2} \approx 1.25\sqrt{r}$ operations if g has order r. All complexity statements in this paper rely on heuristic assumptions; for steps toward a rigorous analysis of the kangaroo method please see Montenegro and Tetali [19].

Gaudry and Schost [14] (building on earlier work of Gaudry and Harley [13]) presented a different approach to solve this problem using a birthday paradox style analysis. Whilst their algorithm is not as fast as that of van Oorschot and Wiener, it is easily parallelisable and importantly there is no requirement to know the number of clients or processors before the algorithm begins. Parallelising the Gaudry-Schost algorithm gives a linear speedup and this also applies to all the algorithms in this paper. For brevity we state all running times for the serial case. The average expected running time of their algorithm is $2.08\sqrt{N}$ group operations on a serial computer (the algorithm of Gaudry and Harley [13] is less efficient). We present their algorithm and recall the analysis of its complexity in Section 2.

Gallant, Lambert and Vanstone [11] and Wiener and Zuccherato [29] showed that the Pollard rho method can be used with equivalence classes (orbits of group elements under an fast computable group homomorphism) to achieve a constant speedup in some groups. In particular, for elliptic curves the rho algorithm can be sped-up by a factor of $\sqrt{2}$ using the equivalence class $\{u, u^{-1}\}$ where u is a group element (which is more commonly written as $\{P, -P\}$ in the case of elliptic curves). In practice, the running times are not so good, since the algorithms use pseudorandom walks which do not behave exactly like true random walks (in particular, walks can fall into short cycles and hence never arrive at a distinguished point; these are called “fruitless cycles” and have been analysed by Duursma, Gaudry and Morain [6] and Bos, Kleinjung and Lenstra [2]).

It seems to be impossible to combine the standard kangaroo method with equivalence classes in general (Section 19.6.3 of [5] claims it can be done but gives no details, and this seems to be an error). Hence, it is necessary to consider other algorithms. A natural observation is that, for a DLP instance (g, h) in an interval of even length N, one can set $h' = hg^{-N/2}$ and then solve $h' = g^n$ where $-N/2 \leq n \leq N/2$. If the discrete logarithm of u lies in the interval $[-N/2, N/2]$ then the equivalence class $\{u, u^{-1}\}$ does correspond to a pair of group elements in the region of interest.

Very recently Pollard [25] developed two new variants of the kangaroo method which require inversion of just one or two group elements. Pollard’s three and four kangaroo variants have heuristic running times of roughly $1.82\sqrt{N}$ and $1.71\sqrt{N}$ group operations respectively. More details about these algorithms will appear in forthcoming joint work.

In Sections 3 and 4 we show how to speed up the Gaudry-Schost method in groups with fast inversion (such as elliptic curves, tori, LUC and XTR). Here fast inversion means that computing u^{-1} for any u in the group is much faster than a general group operation. We also present a further speedup by modifying the search region. Our main result is a method to solve the DLP in an interval in approximately $1.36\sqrt{N}$ group operations. The result uses a new variant of the birthday paradox which is developed in [10]. The theoretical analysis of the algorithm assumes it is run using a truly random walk. In practice one implements the algorithm using a pseudorandom walk which has a number of undesirable consequences, in particular the existence of fruitless cycles. In Section 5 we present experimental results which give a better idea of the actual performance in practice (though it is likely that these figures can be improved).

Our algorithm, as with Gaudry-Schost, requires low storage and can be parallelised with linear speedup very easily.

We indicate in Appendix B how to speed up the Gaudry-Schost algorithm for the multi-dimensional DLP using equivalence classes. A precise analysis of the algorithms in Appendix B is currently an open problem.

2 The Gaudry-Schost Algorithm

To introduce notation and the central ideas, we recall the Gaudry-Schost algorithm [14]. The basic idea is the same as the kangaroo algorithm of Pollard in the van Oorschot and Wiener [22] formulation. Let g and h be the DLP instance we wish to solve, with $h = g^n$ for some integer $-N/2 \leq n \leq N/2$. We run a large number of pseudorandom walks (possibly distributed over a large number of processors). Half the walks are "tame walks", which means that every element in the walk is of the form g^a where the integer a is known. The other half are "wild walks", which means that every element is of the form hg^a where the integer a is known. As is typical in this subject, we visualise the group in terms of the 'exponent space'. More precisely, define the 'tame set'

$$T = [-N/2, N/2]$$

(where by $[N_1, N_2]$ we mean $\{a \in \mathbb{Z} : N_1 \leq a \leq N_2\}$) and the 'wild set'

$$W = n + T = \{n + a : a \in [-N/2, N/2]\}.$$

Although T and W are of the same size, W is a translation of T and of course we do not know the value of n. A tame walk is a sequence of points g^a where $a \in T$ and a wild walk is a sequence of points $g^b = hg^a$ with $b \in W$.

Each walk proceeds until a distinguished point is hit. This distinguished point is then stored on a server, together with the corresponding exponent a and a flag indicating which sort of walk it was. This data is analogous to the 'trap' set in the standard Pollard kangaroo method [24, 26]. When the same distinguished point is visited by two different types of walk we have the “tame-wild collision” $g^{a_1} = hg^{a_2}$ and one solves the DLP. We stress that the algorithm continues until the DLP is solved. Hence the probability of success is 1.

The main difference between the Gaudry-Schost algorithm and the kangaroo algorithm is that when a distinguished point is hit, Gaudry and Schost restart the walk from a random starting point in a certain range, whereas the kangaroos keep on running. The theoretical analysis is different too: Gaudry and Schost use a variant of the birthday paradox whereas Pollard and van Oorschot and Wiener use a different probabilistic argument (see Appendix A).

2.1 Theoretical Analysis

We now recall the precise analysis of the idealised version (i.e., using a truly random walk, rather than a pseudorandom walk) of the Gaudry-Schost algorithm [14]. Gaudry and Schost use the following variant of the birthday paradox, which we will call the Tame-Wild birthday paradox.

Theorem 1. *When sampling uniformly at random from a set of size $R \in \mathbb{N}$, with replacement, and alternately recording the element selected in 2 different lists then the expected number of selections that need to be made in total before we have a coincidence between the lists is $\sqrt{\pi R} + O(1)$.*

Proofs of this theorem can be found in Selivanov [28] or in [27] (which derives it from a result of Nishimura and Sibuya [20]). For simplicity we will omit the $O(1)$ term from all subsequent running times.

Since tame points lie in T and wild points lie in W a collision between tame and wild points can only occur in $T \cap W$. We call such a collision ‘tame-wild’ and this is analogous to a coincidence between the lists in Theorem 1, so we apply Theorem 1 in the case $R = |T \cap W|$.

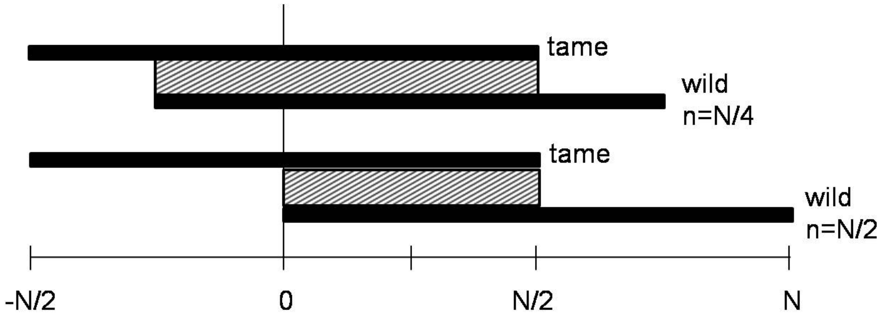

Fig. 1. Overlap between T and W. The sets T and W are represented by black horizontal bars and the shading between them shows the length of the overlap. The first case is when $n = N/4$ and the second case is $n = N/2$.

Figure 1 presents $T \cap W$ in two cases. The first case is $h = g^n$ for $n = N/4$ so $|T \cap W| = 3N/4$ (this is the 'average case'). The second case is $h = g^n$ for $n = N/2$ so $|T \cap W| = N/2$ (which is the 'worst case').

Theorem 2. *(Gaudry-Schost [14]) Let notation be as above. If elements are sampled uniformly at random with replacement alternately from T and W and recorded, the expectation, over all problem instances, of the number of selections before a tame-wild collision is $2.08\sqrt{N}$.*

Proof. The running time of the Gaudry-Schost algorithm is dependent on the problem instance $h = g^n$ but, by symmetry, we can restrict to the case $0 \le n \le N/2$. We write this as $h = g^{xN}$ where $x \in [0, 1/2]$.

Let $R = |T \cap W| = N(1 - x)$. By Theorem 1 we expect to need to sample $\sqrt{\pi R}$ elements (half of each type) of $T \cap W$ to find a collision. To select $\frac{1}{2}\sqrt{\pi R}$ elements in $T \cap W$ when sampling uniformly from T requires selecting

$$\frac{|T|}{|R|}\frac{1}{2}\sqrt{\pi R} = \frac{1}{2}\sqrt{\pi N/(1-x)}.$$

The same argument applies to W. Hence, the expected running time of the algorithm is $\sqrt{\pi N/(1-x)}$ group operations. Note that this is the expected value of the running time, over all choices for the random walk, for a specific problem instance.

We now average this over all problem instances as

$$2\int_0^{1/2}(1-x)^{-1/2}\sqrt{\pi N}dx = 2\sqrt{\pi N}\left[2-\sqrt{2}\right] = 2(2-\sqrt{2})\sqrt{\pi N} \approx 2.08\sqrt{N}.$$

□

This result has been improved to $2.05\sqrt{N}$ by using smaller sets for T and W in [8].

2.2 Pseudorandom Walks and Practical Considerations

Gaudry and Schost present the result in Theorem 2 but they also consider the practical implementation of the algorithm. First, to reduce storage, one does not record every element sampled by the pseudorandom walk but instead uses distinguished points. If we let θ be the probability that an element of the group is a distinguished point then walks are of length $1/\theta$ on average and we require storage of around $\theta\sqrt{N}$ group elements.

Second, it is necessary to use a pseudorandom walk which performs close enough to sampling uniformly at random that the Tame-Wild birthday paradox still applies. Gaudry and Schost, as with the kangaroo method, partition the group into, say, 32 subsets and use a pseudorandom walk where each step is a multiplication of the current group element by g^{a_i}, where a_i is a fixed small positive integer, if the current group element lies in the i-th block of the partition. Our algorithms will necessarily have random walks which step in a "side-to-side" manner, since the equivalence class representative of a group element could be its inverse and steps to the right from the inverse of a group element are the same as steps to the left from the original element. Hence, though we take $a_i \in \mathbb{N}$ and each step is multiplication by g^{a_i}, in practice the walks look like the jumps are of lengths $\pm a_i$. We denote by m the mean of the integers $|a_i|$ and call it the mean absolute step size. For the analysis we recall the following result (note that the mean absolute step size in this walk is $\frac{1}{2}$).

Lemma 1. *(Cofman, Flajolet, Flatto and Hofri [4]) Let $y_0, y_1, \ldots, y_k$ be a symmetric random walk that starts at the origin ($y_0 = 0$) and takes steps uniformly distributed in $[-1, +1]$ then the expected maximum excursion is*

$$E(\max\{|y_i| : 0 \leq i \leq k\}) = \sqrt{\frac{2k}{3\pi}} + O(1)$$

The average 'distance' covered by a random walk, from its starting point to when it hits a distinguished point, is therefore $m/\sqrt{\theta}$. To have good random walks it is essential that this value is sufficiently large so that each walk covers a reasonable proportion of the tame or wild set. If not, then the walks stay very close to their starting point and the probability of two walks colliding is small. On the other hand, when $m/\sqrt{\theta}$ is large then there is a good chance that the pseudorandom walks will sometimes travel outside T or W. Steps outside the regions of interest cannot be included in our probabilistic analysis and so such steps are "wasted". To reduce these wasted steps it is necessary to start walks inside a subset of T and W. More details about how to do this are given in [8].

We therefore state the following heuristic result. The factor $1 + \epsilon$ takes into account the failure of a pseudorandom walk to behave exactly like a random walk, in particular due to effects at the boundaries of the regions.

Heuristic 1 *The average expected running time for the Gaudry-Schost algorithm to solve the DLP in an interval of size N is $2.08(1 + \epsilon)\sqrt{N} + 1/\theta$ group operations for some small $\epsilon > 0$.*

We admit that the statement of Heuristic 1 (and Heuristic 2 later) is essentially vacuous (for example, is $\epsilon = 1$ "small"?). We would like to be able to replace ϵ by $o(1)$. This may be reasonable for Heuristic 1 but it seems unlikely to be reasonable for Heuristic 2. Certainly we feel it is reasonable to suggest that ϵ can be less than 0.1 in both Heuristics 1 and 2.

The standard Gaudry-Schost algorithm is therefore not as fast as the van Oorschot and Wiener version of the Pollard kangaroo method. Nevertheless, we will improve upon their approach in groups with fast inversion, to obtain a method faster than any known method based on kangaroos.

3 Equivalence Classes

Following the work of Gallant, Lambert and Vanstone [11] and Wiener and Zuccherato [29] it is natural to consider a pseudorandom walk on a set of equivalence classes. For the DLP in an interval this only seems to give an improvement when the equivalence class is a set of group elements all of whose discrete logarithms lie in the interval. Groups with fast inversion are good candidates for this.

It is necessary to be able to compute a unique representative of the equivalence class so that one can define a deterministic pseudorandom walk on the equivalence classes. For example consider the group of points on an elliptic curve $E : y^2 = x^3 + Ax + B$ over a finite field $\mathbb{F}_q$ where q is an odd prime. If we let $P = (x_P, y_P) \in E(\mathbb{F}_q)$ then the inverse of P is simply $-P = (x_P, -y_P)$. Now we need a rule to define a unique representative for each equivalence class $\{P, -P\}$. A simple rule in this case is: treat the y-coordinate of P as an integer $0 \leq y_P < q$ and let the unique representative be $(x_P, \min\{y_P, q - y_P\})$. The pseudorandom walk is then defined using the unique equivalence class representative.

If we denote elements of the group by their discrete logarithms and order those in the interval $[-N/2, N/2]$, then the two elements in an equivalence class are equidistant from the centre of the interval. A step to the right for one representative of the equivalence class corresponds to a step to the left for the other. Hence, when using equivalence classes there is no way to avoid having side-to-side walks. This is essentially the reason why the standard kangaroo method cannot be used with equivalence classes.

An important issue is that there is a danger of small cycles in the walks. This phenomena was noted by Gallant, Lambert and Vanstone [11] and Wiener and Zuccherato [29]. This can cause the pseudorandom walks to never reach a distinguished point. A method to get around this problem is "collapsing the cycle" which can be found in Gallant, Lambert and Vanstone [11, Section 6]. A detailed analysis of these issues is given by Bos, Kleinjung and Lenstra [2].

It is natural to try to apply the Gaudry-Schost algorithm on equivalence classes to solve the DLP in an interval of size N.

3.1 The Gaudry-Schost Algorithm on Equivalence Classes

We only give a short sketch of the method, since our main result is a further improvement on the basic idea. Recall that we wish to solve $h = g^n$ where $-N/2 \leq n \leq N/2$. We assume that computing h^{-1} for any h in the group is much faster than a general group operation.

The natural approach is to perform random walks in sets of equivalence classes corresponding to the tame and wild sets of the standard Gaudry-Schost method. In other words, it is natural to make the following definition.

Definition 1. *Define the tame and wild sets by*

$$\begin{aligned} T &= \{\{a, -a\} : a \in [-N/2, N/2]\}\,, \\ W &= \{\{n + a, -(n + a)\} : a \in [-N/2, N/2]\}\,. \end{aligned}$$

Note that $|T| = 1 + N/2 \approx N/2$.

As before, our main focus is on $T \cap W$. When $n = 0$ we have $T = W$ and when n is large then $T \cap W$ is only about half the size of T. However, a subtlety which did not arise in the previous case appears: when $n > N/4$ and $a > N/4$ there is only one way an equivalence class $\{n + a, -(n + a)\}$ can arise, but when $|n|$ is small there can be two ways. Specifically, suppose $-N/4 < n < 0$, then the equivalence class $\{n + a, -(n + a)\}$ can arise from a and from $a' = -2n - a$ (for example, if $n = -N/8$ then $a = N/4$ and $a' = 0$ are such that $\{n + a, -(n + a)\} = \{n + a', -(n + a')\}$). This phenomena means that the Gaudry-Schost algorithm samples from the wild set in a *non-uniform* way and this means we cannot apply Theorem 1 to determine the expected running time of the algorithm. We explain these issues more precisely in the next section.

We do not give an analysis of the average case expected number of group operations for this algorithm. In the next section we make a further optimisation which leads to a better algorithm. A full analysis of the improved algorithm is then given.

4 The New Algorithm

We now give an algorithm for the discrete logarithm problem in an interval for groups with efficient inversion. As usual, let N, g and h be given such that $4 \mid N$, $h = g^n$ and $-N/2 \leq n \leq N/2$. The basic idea is to run the Gaudry-Schost algorithm on the set of equivalence classes. A further speedup is given by defining the wild set W to be, in some sense, smaller than the tame set.

Definition 2. *We define the tame and wild sets (as sets of equivalence classes) by*

$$T = \{\{a, -a\} : a \in [-N/2, N/2]\},$$
$$W = \{\{n + a, -(n + a)\} : a \in [-N/4, N/4]\}$$

where, as always, $[N_1, N_2] = \{a \in \mathbb{Z} : N_1 \leq a \leq N_2\}$.

The algorithm is then immediate. One samples from T and W using pseudorandom walks which are well-defined on equivalence classes. When a walk hits a distinguished point then we store the representative of the equivalence class, its discrete logarithm (or the discrete logarithm of the group element divided by h), and the 'type' of the walk. When the same equivalence class is reached by walks of both types then the discrete logarithm problem is solved.

To understand the algorithm it is necessary to consider a 'fundamental domain' for the sets. In other words, we consider sets which are in one-to-one correspondence with the set of equivalence classes. A fundamental domain for T is $\widetilde{T} = [0, N/2]$; it is clear that every pair $\{a, -a\} \in T$ corresponds to exactly one value $a \in [0, N/2]$. One choice of fundamental domain for W is $\{|n| + a \mid a \in [-N/4, N/4]\}$. However, to visualise $T \cap W$ we really want the fundamental domain for W to consist only of positive values, and this is not the case when $|n| < N/4$. Hence, when $|n| < N/4$, we note that the set W in in one-to-one correspondence with the multi-set

$$\begin{aligned} \widetilde{W} &= \{|n| + a : a \in [-|n|, N/4]\} \cup \{-(|n| + a) : a \in [-N/4, -|n|)\} \\ &= [0, |n| + N/4] + [0, N/4 - |n|). \end{aligned} \tag{1}$$

When $|n| < N/4$, sampling uniformly from W corresponds to sampling uniformly from the multi-set $\widetilde{W}$, which in turn corresponds to sampling $a \in [0, |n| + N/4]$ with probability $4/N$ for $0 \leq a < N/4 - |n|$ and probability $2/N$ for $N/4 - |n| \leq a \leq |n| + N/4$. We describe this as saying that there is a 'double density' of walks in the wild set.

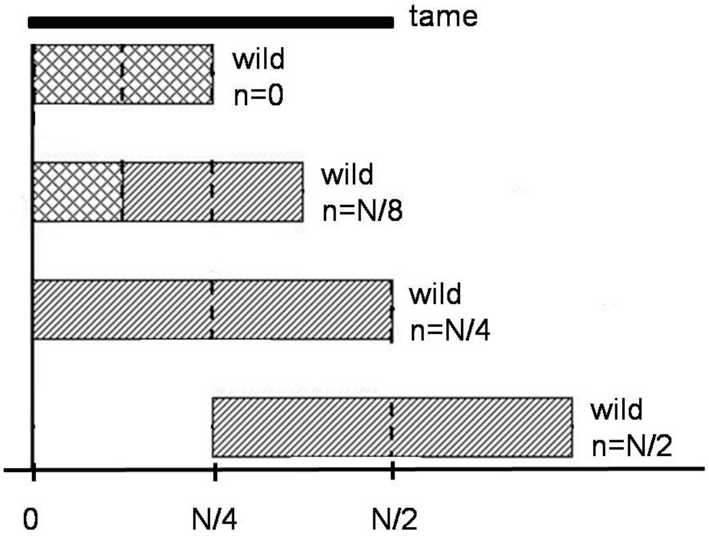

Fig. 2. The set $\widetilde{T}$ is pictured at the top of the diagram as a long black box. The sets $\widetilde{W}$ are given for the values $n = 0, N/8, N/4$ and $N/2$ where the diagonal lines denote single density and the cross hatching denotes double density (i.e., repetitions in the multi-set).

To determine the complexity of the algorithm we need a generalisation of Theorem 1. This is a variant of the birthday paradox which applies to coloured balls and non-uniform probabilities. Such a result is proved in [10].

Theorem 3. *Let $R \in \mathbb{N}$ and $0 \leq A \leq R/2$. Suppose we have an unlimited number of balls of two colours, red and blue, and R urns. Suppose we alternately choose balls of each colour and put them in random urns. Red balls are assigned uniformly and independently to the urns. Blue balls are assigned to the urns independently with the following probabilities: urns $1 \leq u < A$ are used with probability $2/R$, urns $A \leq u \leq R-A$ are used with probability $1/R$, and urns $R-A < u \leq R$ are used with probability 0. Then the expected number of assignments that need to be made in total before we have an urn containing two balls of the same colour is $\sqrt{\pi R} + O(R^{1/4})$.*

We refer to [10] for the proof. However, it is relatively easy to see why the result should be true: The probability that a red ball and a blue ball fall in the same urn is

$$A\tfrac{1}{R}\tfrac{2}{R} + (R-2A)\tfrac{1}{R}\tfrac{1}{R} + A\tfrac{1}{R}0 = \tfrac{1}{R}$$

which is exactly the same as the probability in the case where both red and blue balls are distributed uniformly. One significant difference from the standard Tame-Wild birthday paradox is that there is an increased chance of two or more blue balls being placed in the same urn (and this has the effect of lowering the probability of a collision among balls of different colour). Hence, Theorem 3 does not seem to be an immediate consequence of the results in [20, 28].

Theorem 4. *If elements are sampled uniformly at random with replacement alternately from the sets T and W of Definition 2 and recorded, the expectation, over all problem instances, of the number of selections before a tame-wild collision is*

$$(5\sqrt{2}/4 - 1)\sqrt{\pi N} \approx 1.36\sqrt{N}.$$

Proof. Let $h = g^{xN}$ for $-1/2 \leq x \leq 1/2$. Due to symmetry we only need to look at the positive half of the interval of exponents. As we have seen, when $0 \leq x < 1/4$ we have $W \subseteq T$ and we are sampling in $T \cap W$ uniformly with the tame elements and non-uniformly with the wild elements. On the other hand, when $1/4 \leq x \leq 1/2$ then T and W are both sampled uniformly, but $T \cap W$ is now a proper subset of T and W in general. The analysis therefore breaks into two cases.

In the case $0 \leq x < 1/4$, by Theorem 3 (taking R to be the size of the fundemental domain for T, which is $N/2$), the expected number of group operations to get a collision is $\sqrt{\pi N/2}$.

In the case $1/4 \leq x \leq 1/2$ one sees that $|T \cap W| = 3N/4 - xN = N(3/4 - x)$ (here by $|T \cap W|$ we mean the number of equivalence classes in the intersection) and we are in a very similar situation to the proof of Theorem 2. We need to sample $\sqrt{\pi|T \cap W|}$ points in $T \cap W$ (half of them tame and half wild). Since $|T| = |W| = N/2$ we expect to sample

$$\frac{|T|}{|T \cap W|}\sqrt{\pi|T \cap W|} = \frac{N/2}{N(3/4 - x)}\sqrt{\pi N(3/4 - x)} = \tfrac{1}{2}\sqrt{\pi N/(3/4 - x)}$$

group elements in total.

We now average over all problem instances to get an average case running time.

$$\tfrac{1}{2}\sqrt{\pi N/2} + 2\int_{1/4}^{1/2} \tfrac{1}{2}\sqrt{\pi N/(3/4 - x)} = \sqrt{\pi N}\left(\tfrac{5}{4}\sqrt{2} - 1\right).$$

□

This result suggests the following heuristic statement about the running time of the algorithm using pseudorandom walks. The value ϵ takes into account various undesirable properties of the pseudorandom walk, such as irregular probability distributions at the boundaries of the regions and detecting and escaping from fruitless cycles.

Heuristic 2 *Our algorithm to solve the DLP in an interval of size N in a group with fast inversion has everage case expected running time of approximately* $1.36(1+\epsilon)\sqrt{N} + 1/\theta$ *group operations for some small* $\epsilon > 0$.

As mentioned earlier, we believe that ϵ can be taken to be less than 0.1 and our experimental results suggest this is reasonable.

This is a significant improvement on the standard Gaudry-Schost algorithm (Heuristic 2) and the Improved Pollard kangaroo method with heuristic running time $1.71\sqrt{N}$ group operations [9].

5 Experimental Results

We implemented the Improved Gaudry-Schost algorithm using equivalence classes for solving the DLP in an interval using the software package Magma. The group used was the group of points on the following elliptic curve

$$E : y^2 = x^3 + 40x + 1 \text{ over } \mathbb{F}_p$$

where $p = 3645540875029913$. The group of points has cardinality

$$\#E(\mathbb{F}_p) = 3645540854261153 > 2^{51}.$$

We picked various interval sizes and ran a number of experiments on those intervals. Each experiment involved choosing uniformly at random $-N/2 \leq n \leq N/2$ and solving the DLP for $Q = [n]P$. We counted the number of group operations performed and averaged this over the total number of trials. Walks were not permitted to start within a distance $m\sqrt{2/3\pi\theta}$ from the edge of any of the sets (this is roughly half the size of the expected maximum distance travelled by a walk).

The average number of group operations performed for the different experiments are given in Table 1.

To detect small cycles we stored the previous 30 group elements in the walk in the case $N \approx 2^{34}$ (respectively, 30, 45 group elements for $N \approx 2^{40}, 2^{48}$). Each new step in the walk was compared with the previous 30 (respectively, 35, 45) group elements visited. If this group element had already been visited then the walk is in a cycle. We used a deterministic method to jump out of the cycle (using a jump of distinct length from the other jumps used in the pseudorandom walk) so that the pseudorandom walk as a whole remained deterministic. The cost of searching the list of previous group elements is not included in our experimental results, but our count of group operations does include the "wasted" steps from being in a cycle.

We terminated walks which ran for $5/\theta$ steps without arriving at a distinguished point (the usual recommendation is $20/\theta$ steps; see [22]). This will give us a slightly worse running time than optimal.

There is plenty of room for improvement in these experimental results. First, techniques like those in [2] to handle cycles should lead to improved running

	# of Experiments	Improved GS on equivalence classes
Experiment 1 $N \approx 2^{34}$ $m = 2^{8}$ $\theta = 2^{-5}$	1000	$1.49\sqrt{N}$
Experiment 2 $N \approx 2^{40}$ $m = 2^{11}$ $\theta = 2^{-5}$	300	$1.47\sqrt{N}$
Experiment 3 $N \approx 2^{48}$ $m = 2^{14.5}$ $\theta = 2^{-6}$	50	$1.46\sqrt{N}$

Table 1. Average number of group operations performed by our algorithm for different values of N.

times (though note that we cannot use doublings/squarings when working in an interval). Second, the relationship between the values of m and θ is probably not optimal. Third, one might get better results by not running the same number of tame walks as wild walks or by slightly changing the sizes of the tame and wild regions.

6 Conclusion

We have presented the first algorithm to exploit equivalence classes for the discrete logarithm problem in an interval. Our algorithm can be applied in groups where we have fast inversion such as in the group of points on an elliptic curve. The average expected running time of our algorithm is close to $1.36\sqrt{N}$ group operations. Our practical experiments confirm that we can achieve a significant improvement over previous methods.

Acknowledgements

We thank the anonymous referees of PKC 2010 and ASIACRYPT 2009 for a number of very helpful comments.

References

1. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In Kilian, J., ed.: TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer (2005)

2. Bos, J.W., Kleinjung, T., Lenstra, A.K.: On the use of the negation map in the Pollard Rho method. preprint (2010)
3. Cheon, J.H.: Security Analysis of the Strong Diffie-Hellman Problem. In Vaudenay, S., ed.: EUROCRYPT 2006. LNCS, vol. 4004, pp. 1–11, Springer-Verlag (2006)
4. Cofman, E.G., Flajolet, P., Flatto, L., Hofri, M.: The Maximum of a Random Walk and its Application to Rectangle Packing. Technical report, INRIA (1997)
5. Cohen, H., Frey, G.: Handbook of Elliptic and Hyperelliptic Curve Cryptography. Discrete Mathematics and its Applications. Chapman & Hall/CRC (2005)
6. Duursma, I.M., Gaudry, P., Morain, F.: Speeding up the discrete log computation on curves with automorphisms. In Lam, K.Y., Okamoto E., Xing, C. ed.: ASIACRYPT 1999. LNCS, vol. 1716, pp. 103–121. Springer-Verlag (1999)
7. Galbraith, S.D., Lin, X., Scott, M.: Endomorphisms for Faster Elliptic Curve Cryptography on a Large Class of Curves. In Joux, A., ed.: EUROCRYPT 2009. LNCS, vol. 5479, pp. 518–535. Springer-Verlag (2009)
8. Galbraith, S.D., Ruprai, R.S.: An improvement to the Gaudry-Schost algorithm for multidimensional discrete logarithm problems. In Parker, M.G., ed.: Cryptography and Coding, 12th IMA International Conference. LNCS, vol. 5921, pp. 368–382. Springer (2009)
9. Galbraith, S.D., Pollard, J.M., Ruprai, R.S.: Improving kangaroo and Gaudry-Schost methods for solving the DLP in an interval. In preparation (2010)
10. Galbraith, S.D., Holmes, M.: A non-uniform birthday problem with applications to discrete logarithms. In preparation (2010)
11. Gallant, R., Lambert, R., Vanstone, S.: Improving the Parallelized Pollard Lambda Search on Binary Anomalous Curves. Mathematics of Computation **69**, 1699–1705 (2000)
12. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms. In Kilian, J., ed.: CRYPTO 2001. LNCS vol. 2139, pp. 190–200. Springer-Verlag (2001)
13. Gaudry, P., Harley, R.: Counting Points on Hyperelliptic Curves over Finite Fields. In Bosma, W., ed.: Proceedings of Algorithm Number Theory Symposium - ANTS IV. LNCS, vol. 1838, pp. 313–332. Springer-Verlag (2000)
14. Gaudry, P., Schost, E.: A low-memory parallel version of Matsuo, Chao and Tsujii's algorithm. In Buell, D.A., ed.: Proceedings of Algorithm Number Theory Symposium - ANTS VI. LNCS, vol. 3076, pp. 208–222. Springer-Verlag (2004)
15. Gennaro, R.: An Improved Pseudo-random Generator Based on Discrete Log. In Bellare, M., ed.: CRYPTO 2000. LNCS, vol. 1880, pp. 469–481. Springer-Verlag (2000)
16. Gopalakrishnan, K., Thériault, N., Yao, C.Z.: Solving Discrete Logarithms from Partial Knowledge of the Key. In Srinathan, K., Rangan, C.P., Yung, M., eds.: INDOCRYPT 2007. LNCS, vol. 4859, pp. 224–237. Springer-Verlag (2007)
17. Jao, D., Yoshida, K.: Boneh-Boyen signatures and the Strong Diffie-Hellman problem. In Shacham, H., Waters, B., eds.: Pairing 2009. LNCS, vol. 5671, pp. 1–16. Springer-Verlag (2009)
18. Lim, C.H., Lee, P.J.: A Key Recovery Attack on Discrete Log-based Schemes Using a Prime Order Subgroup. In Kaliski Jr., B.S., ed.: CRYPTO 1997. LNCS, vol. 1294, pp. 249–263. Springer-Verlag (1997)
19. Montenegro, R., Tetali, P.: How long does it take to catch a wild kangaroo? In: 41st ACM Symposium on Theory of Computing. (2009)
20. Nishimura, K., Sibuya, M.: Probability to meet in the middle. Journal of Cryptology **2**, 13–22 (1990)

21. van Oorschot, P.C., Wiener, M.J.: On Diffie-Hellman Key Agreement with Short Exponents. In Maurer, U., ed.: EUROCRYPT 1996. LNCS, vol. 1070, pp. 332–343. Springer-Verlag (1996)
22. van Oorschot, P.C., Wiener, M.J.: Parallel collision Search with Cryptanalytic Applications. Journal of Cryptology **12**, 1–28 (1999)
23. Patel, S., Sundaram, G.: An Efficient Discrete Log Pseudo Random Generator. In Krawczyk, H., ed.: CRYPTO 1998. LNCS, vol. 1462, pp. 304–317. Springer-Verlag (1998)
24. Pollard, J.M.: Monte Carlo Methods for Index Computation mod p. Mathematics of Computation **32**(143), 918–924 (1978)
25. Pollard, J.M.: Three kangaroos are better than two! Private Communication (2009)
26. Pollard, J.M.: Kangaroos, Monopoly and Discrete Logarithms. Journal of Cryptology **13**, 437–447 (2000)
27. Ruprai, R.S.: An improvement to the Gaudry-Schost algorithm for multidimensional discrete logarithm problems and applications. PhD Thesis, Royal Holloway, University of London (2010)
28. Selivanov, B.I.: On waiting time in the scheme of random allocation of coloured particles. Discrete Math. Appl. **5**(1), 73–82 (1995)
29. Wiener, M.J., Zuccerato, R.J.: Faster Attacks on Elliptic Curve Cryptosystems. In Tavares, S.E., Meijer, H., eds.: Selected Areas in Cryptography. LNCS, vol. 1556, pp. 190–200. Springer-Verlag (1998)

A Background on the Pollard kangaroo Method

We first briefly recall the Pollard kangaroo method using distinguished points as described by van Oorschot and Wiener [22] and Pollard [26]. To fix notation: We are given g, h and N and asked to find $0 \leq n \leq N$ such that $h = g^n$.

As with the rho method, the kangaroo method relies on a pseudorandom walk, however steps in the kangaroo walk correspond to known small increments in the exponent (in other words, kangaroos make small jumps). The tame kangaroo starts in the middle of the interval (i.e., at $g^{N/2}$) and jumps towards the right. The wild kangaroo starts at the group element h and jumps to the right using the same pseudorandom walk. On a serial computer one alternately jumps the tame and wild kangaroos. Every now and then a tame or wild kangaroo lands on a distinguished group element u and stores it in a sorted list, binary tree or hash table together with its discrete logarithm (if the kangaroo is tame) or the discrete logarithm of uh^{-1} (if the kangaroo is wild). Once the same group element is visited twice by different kangaroos the DLP is solved.

The kangaroo method is not analysed using the birthday paradox but using the mean step size m of the pseudorandom walks. Once the rear kangaroo reaches the starting point of the front kangaroo it is jumping over a region where roughly one in m group elements have been visited by the front kangaroo. Hence, there is a roughly $1/m$ probability at each step that the back kangaroo lands on a footprint of the front kangaroo. Therefore, the walks collide after an expected m steps.

One obtains the heuristic average case expected running time of approximately $2\sqrt{N}$ group operations as follows: Choose $m = \sqrt{N}/2$. The rear kangaroo is, on average, distance $N/4$ from the front kangaroo. The rear kangaroo therefore performs $N/(4m)$ jumps to reach the starting point of the front kangaroo, followed by m more steps until the walks collide (and then a small number more steps until a distinguished point is hit). Since there are two kangaroos in action the total running time is roughly $2(N/(4m) + m) = 2\sqrt{N}$ group operations.

B Two-Dimensional Problems

One can consider the multi-dimensional DLP: Given $g_1, \ldots, g_d, h$ and bounds $N_1, \ldots, N_d$, to compute $n_1, \ldots, n_d \in \mathbb{Z}$ such that $h = g_1^{n_1} \cdots g_d^{n_d}$ and $|n_i| \leq N_i$ for $1 \leq i \leq d$. We call the integer d the dimension. The size of the solution region is $N = \prod_{i=1}^{d}(2N_i + 1)$. This problem arises in a number of applications. For example, Gaudry and Schost [14] use algorithms for the 2-dimensional DLP in point counting on hyperelliptic curves of genus 2.

The 2-dimensional DLP also arises if one tries to analyse the security of elliptic curve cryptography using the Gallant-Lambert-Vanstone (GLV) method [12]. In this method one has an efficiently computable group homomorphism ψ and one computes nP for $P \in E(\mathbb{F}_q)$ and $n \in \mathbb{N}$ as $n_1 P + n_2 \psi(P)$ where $|n_1|, |n_2| \approx \sqrt{n}$. There is an algorithm to compute the pair (n_1, n_2) from n, but a natural trick is to choose (n_1, n_2) directly. It is tempting to choose $|n_1|$ and $|n_2|$ to be a little smaller than $\sqrt{n}$, and the extent to which this can be done without losing security depends on the difficulty of the 2-dimensional DLP. Gaudry and Schost do not give a precise figure for the running time of this algorithm but we have the following heuristic under the usual assumptions (see [8] for further details of this result and an improvement of the constant from 2.43 to 2.36).

Heuristic 3 *The Gaudry and Schost [14] algorithm solves the 2-dimensional DLP in as above in* $2.43(1 + \epsilon)\sqrt{N} + 1/\theta$ *group operations for small* $\epsilon > 0$.

B.1 Solving Using Equivalence Classes

In groups with efficiently computable inverse (such as the groups of interest to Gaudry and Schost and the GLV method) one can consider equivalence classes as we did in the 1-dimensional case. To be precise, let

$$T = \{\{(x, y), (-x, -y)\} : x, y \in \mathbb{Z}\ ,\ -N_1 \leq x \leq N_1, -N_2 \leq y \leq N_2\}$$

be the set of equivalence classes of points in a box of area $N = (2N_1 + 1)(2N_2 + 1)$ centered at 0. For (n_1, n_2) such that $-N_i \leq n_i \leq N_i$ $(i \in \{1, 2\})$ we consider the set

$$W = \left\{ \{(n_1 + u_1, n_2 + u_2), (-(n_1 + u_1), -(n_2 + u_2))\} : \begin{array}{r} u_1, u_2 \in \mathbb{Z}, \\ -N_1/2 \leq u_1 \leq N_1/2, \\ -N_2/2 \leq u_2 \leq N_2/2 \end{array} \right\}.$$

To analyse the algorithm again requires visualising the sets via a 'fundamental domain'. Since the map $(x, y) \mapsto (-x, -y)$ is rotation by 180 degrees, a natural fundamental domain is the halfplane $y \geq -x$. One therefore defines the fundamental domain $\widetilde{T}$ for T to be

$$\widetilde{T} = \{(x, y) : -N_1 \leq x \leq N_1, -x \leq y \leq N_2\}.$$

Note that $|\widetilde{T}| \approx 2N_1N_2$. A fundamental domain for $\widetilde{W}$ which is contained in $\widetilde{T}$ is easily defined, but note that, as in Section 4, this can be a multi-set and we can again be in the case of non-uniform sampling of $\widetilde{W}$. Indeed, when $0 \leq n_1 < N_1/2$ and $0 \leq n_2 < N_2/2$ this is the case and the region which has 'double density' has area $A = \frac{1}{2}(N_1 - 2n_1)(N_2 - 2n_2)$. When $n_1 \geq N_1/2$ or $n_2 \geq N_2/2$ then the distribution on $\widetilde{W}$ is uniform but $\widetilde{W}$ is not usually contained in $\widetilde{T}$ any more. In these cases $|\widetilde{T} \cap \widetilde{W}|$ varies between $N_1N_2 = \frac{1}{2}|\widetilde{T}|$ and $\frac{1}{4}N_1N_2 = \frac{1}{8}|\widetilde{T}|$.

We considered an algorithm which chooses elements from T and W uniformly at random, selecting elements with a ratio of $2 : 1$ (i.e., twice as many tame walks as wild walks, since the tame set is at least twice as big as the wild set).

Our rough calculations suggest that the algorithm (when using a truly random walk) should require fewer than $2.01\sqrt{N}$ group operations. This is a significant speedup over the algorithm of Gaudry and Schost for the cases of practical interest. It remains an open problem to find optimal parameters for this algorithm, to analyse its complexity precisely, and to give experimental results which show how closely one can get to the idealised theoretical analysis.

B.2 Larger Equivalence Classes in the GLV Method

We now assume that $N_1 = N_2$ and that the 2-dimensional DLP of interest is $Q = n_1P + n_2\psi(P)$ with $|n_1|, |n_2| \leq N_1$. Again write $N = (2N_1 + 1)(2N_2 + 1)$. Since one knows the logarithm of $\psi(P)$ to the base P it is sufficient to compute n_1 and n_2.

Frequently with the GLV method [12] the homomorphism ψ satisfies $\psi^2 = -1$. This happens, for example, with the standard curve $y^2 = x^3 + Ax$ over $\mathbb{F}_p$ with $\psi(x, y) = (-x, iy)$. It also holds for the homomorphisms used by Galbraith, Lin and Scott [7]. In this setting one can consider the equivalence classes

$$\{Q, -Q, \psi(Q), -\psi(Q)\}$$

of size 4. If $Q = n_1P + n_2\psi(P)$ then these 4 points correspond to the pairs of exponents

$$\{(n_1, n_2), (-n_1, -n_2), (-n_2, n_1), (n_2, -n_1)\}$$

and so action by ψ corresponds to rotation by 90 degrees.

It is natural to apply the Gaudry-Schost algorithm on these equivalence classes. We take the sets T and W analogous to those in Section B.1. Finding a suitable fundamental domain for the symmetry under rotation is not hard (for example take $\{(x, y) : 0 \leq x, 0 \leq y\}$). One now finds that some regions of the wild set can have quadruple density (as well as single and double density).

Again, it remains an open problem to determine the optimal algorithm for this problem and to estimate its complexity. A very rough calculation suggests that there is an algorithm for this problem (using a truly random walk) which requires fewer than $1.11\sqrt{N}$ group operations.