An Improvement to the Gaudry-Schost Algorithm for Multidimensional Discrete Logarithm Problems

Steven 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, UK
r.s.ruprai@rhul.ac.uk

Abstract. Gaudry and Schost gave a low-memory algorithm for solving the 2-dimensional discrete logarithm problem. We present an improvement to their algorithm and extend this improvement to the general multidimensional DLP. An important component of the algorithm is a multidimensional pseudorandom walk which we analyse thoroughly in the 1 and 2 dimensional cases as well as giving some discussion for higher dimensions.
Keywords: discrete logarithm problem (DLP)

1 Introduction

The discrete logarithm problem (DLP) is: Given a cyclic group G (in additive notation) of order r and $P, Q \in G$ where P is a generator of the group, find a positive integer n such that $Q = [n]P$. This is an important computational problem due to applications in public key cryptography. In a generic group there are standard algorithms for solving the DLP such as Baby-Step Giant-Step (BSGS) [18] (see, for example, [19]) and Pollard rho [14]. Van Oorschot and Wiener [21, 22] give a version of Pollard Rho, suitable for distributed computing, based on the idea of distinguished points. For further works on Pollard rho see [3, 20]. BSGS solves the DLP in $O(\sqrt{r})$ group operations but also requires $O(\sqrt{r})$ group elements of storage. Pollard rho and van Oorschot-Wiener also solve the DLP in heuristic *expected* $O(\sqrt{r})$ group operations but require only constant storage. We also refer to [10] for a rigorous analysis of the Pollard rho algorithm.

A higher dimensional version of the DLP arises in a number of applications (see Section 6). We now give a precise definition of it.

Definition 1 (Multidimensional DLP). *Let G be an abelian group and let $P_1, P_2, \ldots, P_d, Q \in G$ and $N_1, \ldots N_d \in \mathbb{N}$ be given. The d-dimensional discrete logarithm problem is to find (if they exist) integers $n_i \in [-N_i, N_i]$, for $1 \leq i \leq d$, such that*

$$Q = [n_1]P_1 + [n_2]P_2 + \ldots + [n_d]P_d. \tag{1}$$

We call $(n_1, \ldots, n_d)$ the exponent vector *of Q. We write $\tilde{N}_i = 2N_i + 1$ and*

$$N = \prod_{i=1}^{d} \tilde{N}_i. \tag{2}$$

Note that G may or may not be cyclic. In the case of a cyclic group of prime order and $\tilde{N}_i = \#G$ this computational problem is the representation problem introduced by Brands [2] in 1993.

It is easy to adapt the BSGS algorithm to the multidimensional case, as shown by Matsuo et. al. [12], and this algorithm requires $O(\sqrt{N})$ group operations and $O(\sqrt{N})$ group elements of storage. Pollard describes

* This work supported by EPSRC grant EP/D069904/1.
** The work described in this report has in part been supported by the Commission of the European Communities through the ICT program under contract ICT-2007-216676. The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

his kangaroo method in [14, 15] to solve the 1-dimensional case (i.e., $d = 1$) also in *expected* $O(\sqrt{N})$ group operations but requiring only constant storage. Van Oorschot and Wiener [21, 22] also give a version of the kangaroo method which is suitable for distributed computing. It seems unlikely that the kangaroo algorithm can be adapted to the case of dimension $d \geq 2$.

Gaudry and Schost [9] present an algorithm to solve the 2-dimensional case using random walks and distinguished points. Unlike the kangaroo method, their algorithm is analysed using a variant of the Birthday Paradox. They also analyse their algorithm in the 1-dimensional case. We give brief details of their algorithm in Section 2 as well as extending their approach to any dimension. Theorem 2 generalises the result of Gaudry and Schost. Note that the analysis of the Gaudry-Schost algorithm is only heuristic; to be able to easily state theorems we consider an idealised model.

In Section 3 we present a simple idea which improves the Gaudry-Schost algorithm in all dimensions. In Section 4 we give more details about how to put the improved algorithm into practice by correctly choosing parameters. In Section 5 we discuss the implementation issues in higher dimensions and then look at some applications of solving the multidimensional DLP in Section 6.

2 The Gaudry-Schost Algorithm

We first recall the Pollard kangaroo algorithm for the 1-dimensional DLP. In other words, we have $Q = [n_1]P_1$ with $-N_1 \leq n \leq N_1$ (and $N = 2N_1 + 1 \approx 2N_1$). The algorithm finds two integers $a, b \in \mathbb{N}$ such that $[a]P_1 = Q + [b]P_1$ and hence solves the DLP.

The crucial idea is to use a "deterministic pseudorandom walk". More precisely, a selection function $S : G \to \{1, \ldots, n_S\}$ is chosen, which can be interpreted as partitioning the group into n_S sets of roughly equal size. Suitable integers $z_1, \ldots, z_{n_S}$ are chosen (e.g., uniformly chosen from $[0, c_1\sqrt{N}]$ for some constant c_1) and, given $x_1 \in G$, the deterministic pseudorandom walk $x_1, x_2, \cdots \in G$ proceeds as

$$x_{i+1} = x_i + [z_{S(x_i)}]P_1.$$

Pollard uses a *tame walk* which starts at $x_1 = [N_1]P$ and takes $n = O(\sqrt{N})$ steps before stopping at x_n. Pollard also uses a *wild walk* which starts at $y_1 = Q$ and also takes $O(\sqrt{N})$ steps. If the wild walk lands on a footprint of the tame walk then the wild walk follows the path of the tame walk and eventually we find $y_m = x_n$. By storing not just the values x_n and y_m but also the corresponding integers a and b such that $x_n = [a]P_1$ and $y_m = Q + [b]P_1$ we can solve for the DLP.

Van Oorschot and Wiener [21, 22] give a version of this algorithm which is not only distributed but has better expected running time. The idea is to use distinguished points, so instead of just storing a single end-point (x_n, a) one stores a moderate number of intermediate values. Indeed, van Oorschot and Wiener showed that the expected running time is heuristically $2\sqrt{N} + 1/\theta$ group operations for storage requirement approximately $2\theta\sqrt{N}$ group elements (throughout the paper, θ denotes the probability that a group element is a distinguished point).

The Gaudry and Schost algorithm is motivated by the above ideas but the analysis is very different. Before we describe the Gaudry-Schost algorithm we must first define the following sets of exponents.

Definition 2. *Let notation be as in Definition 1 and suppose $Q = [n_1]P_1 + \cdots + [n_d]P_d$ for some integers $n_i \in [-N_i, N_i]$. Define the Tame set T and the Wild set W, which are subsets of $\mathbb{Z}^d$, by*

$$T = \left\{(a_1, a_2, \ldots, a_d) \in \mathbb{Z}^d \ : \ a_i \in [-N_i, N_i] \text{ for all } 1 \leq i \leq d\right\},$$
$$W = (n_1, n_2, \ldots, n_d) + T = \{(n_1 + a_1, \ldots, n_d + a_d) \ : \ (a_1, \ldots, a_d) \in T\}.$$

The exponent vector of Q lies somewhere in the tame set T. The wild set W is a translation of T which is centred on the exponent vector of Q. The sets T and W are orthotopes in $\mathbb{Z}^d$ (i.e., d-dimensional products of intervals).

The basic idea of the Gaudry-Schost algorithm [9] is the same as the kangaroo algorithm of Pollard in the van Oorschot and Wiener formulation. Let the multidimensional DLP problem instance of dimension d

be given as in Definition 1. 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 $[a_1]P_1 + [a_2]P_2 + \ldots + [a_d]P_d$ where the integer tuple $(a_1, a_2, \ldots, a_d) \in T$ (though note that with very small probability some walks will go outside T). The other half are "wild walks", which means that every element is of the form $Q + [b_1]P_1 + [b_2]P_2 + \ldots + [b_d]P_d$ where the integer tuple $(b_1, b_2, \ldots, b_d) \in T$. Each walk proceeds until a distinguished point is hit. This distinguished point is then stored in an easily searched structure (e.g., a binary tree), together with the corresponding d-tuple of exponents. We maintain two such structures: one to store the points found by tame walks and one for the wild walks. When the same distinguished point is visited by two different types of walk we have the collision $[a_1]P_1+[a_2]P_2+\ldots+[a_d]P_d = Q + [b_1]P_1 + [b_2]P_2 + \ldots + [b_d]P_d$ and one solves the multidimensional DLP as follows

$$Q = [a_1 - b_1]P_1 + [a_2 - b_2]P_2 + \ldots + [a_d - b_d]P_d.$$

A significant 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 the appropriate set, 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 (based on the mean step size).

We now present the theoretical analysis of the Gaudry-Schost algorithm. Our main result in this section is Theorem 2 which generalises the results of Section 3.1 and 4.1 of [9]. We first recall a tool from probability theory which we will need (this result was also used by Gaudry and Schost) and which we call the *Tame-Wild Birthday Paradox*.

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

Proof. See Nishimura and Sibuya [13] or [17]. □

Since tame walks lie in T (with high probability) and wild walks lie in W, a collision between tame and wild walks can only occur in $T \cap W$. We call this set the *overlap* and denote its size by M. We will therefore apply Theorem 1 in $T \cap W$ only. To get a rough idea of the running time of the Gaudry-Schost algorithm we consider an idealised version of it which includes two simplifying assumptions. First, we assume that the points in T and W are chosen uniformly at random, rather than using a pseudorandom walk. Second, we assume that all points are stored (in other words, all points are distinguished) so that a tame-wild collision is detected immediately.

Theorem 2. *Given the multidimensional discrete logarithm problem as described in Definition 1 with N the cardinality of the search space as given by equation (2). Then the expected number of group operations (in the idealised model) in the worst case of the original Gaudry-Schost algorithm is*

$$2^{d/2}\sqrt{\pi N}, \tag{3}$$

and the average case expected number of group operations is

$$((4 - 2\sqrt{2})^d + o(1))\sqrt{\pi N}. \tag{4}$$

When $d = 1$ the worst and average case running times are approximately $2.51\sqrt{N}$ and $2.08\sqrt{N}$ group operations. When $d = 2$ the worst and average case running times are approximately $3.54\sqrt{N}$ and $2.43\sqrt{N}$.

Proof. In the worst case Q lies in a 'corner' of the search space and so the overlap between the original tame and wild sets has cardinality $M = \frac{N}{2^d}$. Therefore we expect only about $1/2^d$ of the points sampled to be in $T \cap W$. The running time is therefore 2^d times the expected number of of elements sampled in $T \cap W$ to

get a collision. Using the Tame-Wild Birthday Paradox given in Theorem 1, the expected number of group operations in the worst case is therefore given by

$$2^d\sqrt{\pi M} = 2^{d/2}\sqrt{\pi N}$$

which proves the result presented in equation (3). To find the average case expected running time we have to average over all possible Q. Without loss of generality let us consider one of the 2^d possible orthonts (i.e., sub-orthotopes corresponding to one corner of the search space). In other words, suppose that

$$Q = [x_1\tilde{N}_1]P_1 + [x_2\tilde{N}_2]P_2 + \ldots + [x_d\tilde{N}_d]P_d$$

with $x_i \in [0, 1/2]$. The cardinality of the overlap between T and W is

$$M = \left(\frac{1}{2} + x_1\right)\tilde{N}_1 \cdot \left(\frac{1}{2} + x_2\right)\tilde{N}_2 \cdots \left(\frac{1}{2} + x_d\right)\tilde{N}_d = \left(\prod_{i=1}^{d}\left(\frac{1}{2} + x_i\right)\right) N.$$

Therefore we expect only about

$$M/N = \prod_{i=1}^{d}\left(\frac{1}{2} + x_i\right)$$

of the walks to be in $T \cap W$. So the average case expected running time is approximately

$$2^d \int_{x_1=0}^{1/2}\int_{x_2=0}^{1/2}\cdots\int_{x_d=0}^{1/2}\prod_{i=1}^{d}\left(\frac{1}{2} + x_i\right)^{-1/2}\sqrt{\pi N}\, dx_1\, dx_2 \ldots dx_d$$

$$= 2^d\sqrt{\pi N}\left(\int_{x=0}^{1/2}\left(\frac{1}{2} + x\right)^{-1/2} dx\right)^d = (4 - 2\sqrt{2})^d\sqrt{\pi N}$$

which proves the result presented in equation (4). □

3 The Improved Gaudry-Schost Algorithm

We now give the main result of the paper, which is a version of the Gaudry-Schost algorithm which has a faster running time. The key observation is that the running time of the Gaudry-Schost algorithm depends on the size of the overlap between the tame and wild sets. If it is possible to make the size of this overlap constant for all possible Q then the expected running time will be constant for all problem instances. We achieve this by choosing walks which only cover certain subsets of T and W.

Precisely, instead of using the sets T and W of the previous section we use an orthotope T' of size $(2/3)^d N$ centered in T and a space W' of the same size but split into 2^d disjoint sets in the 'corners' of W. In the 1-dimensional case we have $T' = [-2N_1/3, 2N_1/3] \cap \mathbb{Z} \subseteq T$ and $W' = ([n_1 - N_1, n_1 - N_1/3] \cup [n_1 + N_1/3, n_1 + N_1]) \cap \mathbb{Z} \subseteq W$ (see Figure 1). The algorithm proceeds in exactly the same way as before.

For the theoretical analysis we again use the idealised model where we ignore the issue of pseudorandom walks and assume that we are storing every element visited.

Theorem 3. *Given the multidimensional discrete logarithm problem as described in Definition 1 with N the cardinality of the search space as given by equation (2). Then the expected number of group operations (in the idealised model) for the improved Gaudry-Schost algorithm is*

$$\left(\frac{2^d}{3^{d/2}} + o(1)\right)\sqrt{\pi N}.$$

This is the expected running time in the best, worst and average cases. When $d = 1$ and $d = 2$ this is approximately $2.05\sqrt{N}$ and $2.36\sqrt{N}$ group operations respectively.

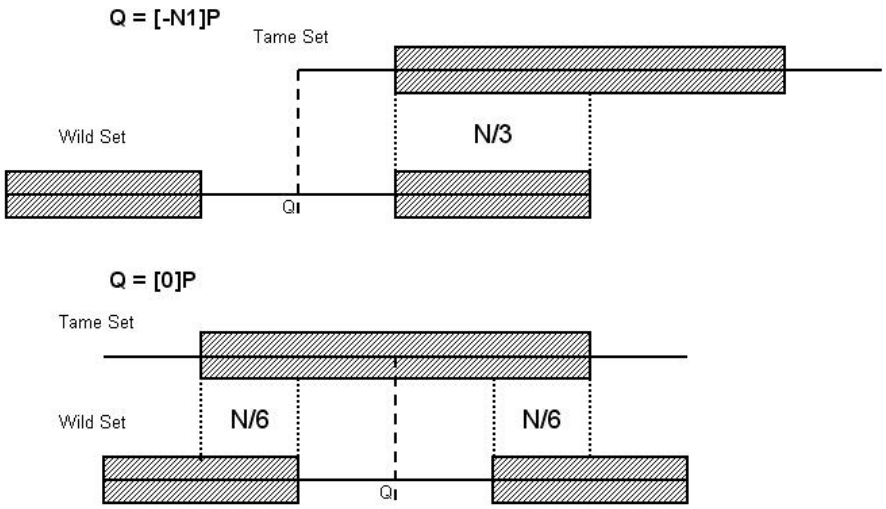

Fig. 1. Depiction of T, W, T' and W' for two extreme cases of Q. The thin horizontal lines denote T and W while the fatter shaded rectangles denote T' and W'.

Proof. We search in an orthotope of size $k^d N$ centred in the middle of the tame set and we search a space of the same size but split into 2^d disjoint sets in the 'corners' of the wild set. Each of these disjoint sets of size $((k/2)^d + o(1))N$. We now compute the volume of the overlap $T' \cap W'$ which is a union of orthotopes. Since the volume of an orthotope is the product of the lengths of its sides our problem reduces to the 1-dimensional case.

It can be easily shown that to make the overlap constant for all problem instances one must take $k = 2/3$ (see Figure 1). Then for all problem instances the cardinality of the overlap is $M' = \frac{N}{3^d}$ and the size of the new search space is given by $N' = \frac{2^d N}{3^d}$. Therefore the proportion of steps in $T' \cap W'$ is approximately

$$\frac{M'}{N'} = \frac{1}{2^d}.$$

Using the Tame-Wild Birthday Paradox analysis, the expected number of group operations before a collision is approximately

$$2^d \sqrt{\pi M'} = 2^d \sqrt{\pi \frac{N}{3^d}} = \frac{2^d}{3^{d/2}} \sqrt{\pi N}.$$

This completes the proof. □

Theorem 3 is an improvement on the average and the worst case expected running time of the Gaudry-Schost algorithm in all dimensions. In the 2-dimensional case the expected running time falls from $2.43\sqrt{N}$ group operations in the original Gaudry-Schost to $2.36\sqrt{N}$ group operations in the improved Gaudry-Schost. In the worst case the improvement is even bigger from $3.54\sqrt{N}$ to $2.36\sqrt{N}$. The benefit of this new approach increases with d.

In the 1-dimensional case the original Gaudry-Schost algorithm was not competitive with the van Oorschot and Wiener variant of the Pollard kangaroo method. Our improvement is also not competitive.

4 Pseudorandom Walks and Counting Bad Points

In this section we consider some problems which arise in practice and the techniques used by Gaudry and Schost to combat them. Firstly we do not want to store every point visited, which is where the idea

of distinguished points comes into play. Denote by θ the probability that an element of the group is a distinguished point. In practice the value of θ is chosen as a tradeoff between the work of each client required to find a distinguished point (approximately $1/\theta$ group operations) and the total storage on the server (proportional to $\theta\sqrt{N}$ group elements). To have an algorithm whose expected number of group elements of storage is constant one would set $\theta = c/\sqrt{N}$ for some constant c.

The more serious assumption in the idealised model is that elements are selected from the tame and wild sets uniformly at random. The Gaudry-Schost algorithm uses a deterministic pseudorandom walk. Our experiments suggest one can choose walks which behave "close enough" (in the sense that the Tame-Wild Birthday Paradox seems to hold) to selecting uniformly at random. In practice, it seems to be impossible to design pseudorandom walks which cover T or W uniformly but which do not sometimes overstep the boundaries. Steps outside the regions of interest cannot be included in our probabilistic analysis and so such steps are "wasted". We call these "type 2" bad points. Another assumption of the idealised model is that tame-wild collisions are always detected. In principle it can happen that a walk takes an exceptionally long time to hit a distinguished point (i.e., that a large number of consecutive steps happen to not be distinguished; this phenomena can also be caused by cycles, but that is not our concern in this section). In the parallelised setting we should assume that individual processors may be relatively slow and so, in principle, it could happen that some processor never finds a distinguished point. Such steps are therefore also wasted and we call them "type 1" bad points.

To guard against bad steps of type 1 van Oorschot and Wiener [22] set a maximum number of steps in a walk. If a processor takes this many steps without hitting a distinguished point then it restarts on a new random starting point. They choose this maximum to be $20/\theta$ steps and show that the proportion of bad points of type 1 is at most 5×10^{-8}. This result applies in all dimensions.

We now consider bad points of type 2. This depends on both the pseudorandom walk and the dimension. The main idea is to start walks in proper subsets of T' and W' and to set up these regions so that there is good chance to cover all of T' and W' but also so that the probability that a walk goes outside T' and W' is relatively small.

4.1 1-Dimensional Case

In the 1-dimensional case we use a standard pseudorandom walk as used by Pollard [15] and van Oorschot and Wiener [22] i.e., small positive jumps in the exponent. The average distance in the exponent traveled by a walk is m/θ, where m is the mean step size and θ is the probability that a point is distinguished. For example, one may have $m = c_1\sqrt{N}$ and $\theta = c_2/\sqrt{N}$ for some constants $0 < c_1 < 1$ and $1 < c_2$. Therefore, to reduce the number of bad steps of type 2 we do not start walks within this distance of the right hand end of the interval. In other words, we do not start walks in the following subset of T'.

$$\left[\frac{2N_1}{3} - \frac{m}{\theta}, \frac{2N_1}{3}\right] \cap \mathbb{Z} \tag{5}$$

The analogous omitted subsets for W are the following

$$\left[n_1 - \frac{N_1}{3} - \frac{m}{\theta}, n_1 - \frac{N_1}{3}\right] \cap \mathbb{Z} \text{ and } \left[n_1 + N_1 - \frac{m}{\theta}, n_1 + N_1\right] \cap \mathbb{Z}. \tag{6}$$

Lemma 1. *Let m be the mean step size and* max *the maximum step size. Let the subsets where walks will not start be given by equations (5) and (6). The probability that a walk has bad points of type 2 is at most*

$$p = \frac{20\max - m}{2N_1\theta/3 - m}. \tag{7}$$

Proof. Let $T'' = [-2N_1/3, 2N_1/3 - m/\theta] \cap \mathbb{Z}$ be the set of possible starting points of tame walks and let W'' be the set of possible starts points of wild walks. One expects there to be at least twice as many wild walks with bad points of type 2 as tame walks. Hence it is sufficient to consider the probability for wild walks only.

Let

$$X = [n_1 + N_1/3, n_1 + N_1 - m/\theta]$$

be one of the components of W''. Note that $\#X = 2N_1/3 - m/\theta$. Since walks travel distance at most $20 \max/\theta$ the only walks which can possibly have bad points of type 2 are ones which start in $[n_1 + N_1 - 20 \max/\theta, n_1 + N_1 - m/\theta]$. Hence the probability that a walk starting in X has bad points of type 2 is

$$\frac{20 \max/\theta - m/\theta}{\#X}.$$

The result follows. □

For the values $m = c_1\sqrt{N_1}$, $\max = 2m$ and $\theta = c_2/\sqrt{N_1}$ the value of p in equation (7) is $39c_1/(2c_2/3 - c_1)$ which can be made arbitrarily small by taking c_2 sufficiently large (i.e., by storing sufficiently many distinguished points). Hence, even making the over-cautious assumption that all points are wasted if a walk contains some bad points of type 2, the expected number of walks in the improved Gaudry-Schost algorithm can be made arbitrarily close to the desired value when N is large. In practice it is reasonable to store at least 2^{30} distinguished points, which is quite sufficient to minimise the number of bad points of type 2.

We stress that the choices of m, max and θ are not completely arbitrary. Indeed, if one wants to bound the number of bad points of type 2 as a certain proportion of the total number of steps (e.g., 1%) then for a specific N there may be limits to how large θ and max can be. For smaller N we cannot have too small a probability of an element being distinguished or too large a mean step size.

4.2 2-Dimensional Case

In the 2-dimensional case, Gaudry and Schost [9] use a walk which goes forwards with respect to one axis and side-to-side in the other. In other words, each step of the walk adds a group element of the form $[a]P_1 + [b]P_2$ where $0 \leq a$ is typically very small (possibly zero sometimes) and $b \in \mathbb{Z}$. Our experience suggests that it is sufficient in practice to use walks of the following form (at least, when $N_1 \approx N_2$ and $N_1 > N_2$).

Definition 3. *Let $S : G \to \{1, \ldots, n_S\}$ partition G. Let m_2 be a parameter (the* mean absolute step size*). Let $z_1, \ldots, z_{n_S} \in \mathbb{Z}$ be chosen uniformly at random in the interval $[-2m_2, 2m_2]$. The pseudorandom walk from a given value $x_i \in G$ is*

$$x_{i+1} = x_i + [1]P_1 + [z_{S(x_i)}]P_2$$

In the P_1 component, as every step is of size 1, the size of the subsets of exponents where walks do not start will be the expected walk length, $\frac{1}{\theta}$, from the right hand edge of the interval of exponents. Since $m = \max = 1$ in this case, one can apply Lemma 1 directly.

Lemma 2. *Let notation be as above. The probability that a walk has bad points of type 2 in the P_1 component is at most*

$$\frac{19}{2N_1\theta/3 - 1}.$$

If $N_1 \approx N_2$ then $\theta = c/N_1$ and this probability is $19/(2c/3 - 1)$ which can be made arbitrarily small by choosing c to be large (i.e., storing many distinguished points).

The analysis of the bad points of type 2 in the P_2 component is different. We use the following result by Cofman, Flajolet, Flatto and Hofri [4] to address this problem (Gaudry and Schost [9] use a similar result in their analysis).

Lemma 3. *Let $y_0, y_1, \ldots, y_n$ be a symmetric random walk that starts at the origin ($y_0 = 0$) and takes steps uniformly distributed in $[-1, +1]$ then the expected value of $\max\{|y_i| : 0 \leq i \leq n\}$ is*

$$\sqrt{\frac{2n}{3\pi}} + O(1).$$

Note that the mean absolute step size in this walk is $\frac{1}{2}$.

We prohibit walks from starting in intervals of length

$$\delta = 2m_2\sqrt{\frac{2}{3\pi\theta}}. \tag{8}$$

at each edge (in the P_2 direction) of the tame and wild regions. To be precise, the region in which tame walks are permitted to start is

$$T'' = ([-2N_1/3, 2N_1/3 - 1/\theta] \times [-2N_2/3 + \delta, 2N_2/3 - \delta]) \cap \mathbb{Z}^2.$$

Note that $\#T'' = (4N_1/3 - 1/\theta)(4N_2/3 - 2\delta)$.

Lemma 4. *Suppose a pseudorandom walk as in Definition 3 is being used. Let m_2 be the mean absolute step size in the P_2 component, and $\max_2$ the maximum absolute step size. Let δ be as in equation (8). The probability that a walk has bad steps of type 2 in the P_2 component is at most*

$$\frac{40\max_2}{(2N_2/3 - 2\delta)\theta}. \tag{9}$$

A sharper version of this result, with a more complicated proof, is given in Lemma 5.2.11 of [17].

Proof. As with the earlier results, it suffices to deal with wild walks. Consider the projection of one component of the wild set W'' onto the P_2 axis, for example

$$X = [N_2/3 + \delta, N_2 - \delta] \cap \mathbb{Z}.$$

Note that $\#X = 2N_2/3 - 2\delta$. Since the maximum distance travelled by a walk is, in absolute value, $20\max_2/\theta$, bad points of type 2 can only arise from walks which start within this distance of the edge (on either side). Hence, the probability that a walk starting in X has bad points of type 2 is at most

$$2\frac{20\max_2/\theta - \delta}{\#X} \leq 2\frac{20\max_2}{\theta\#X}.$$

This gives the result. □

Suppose $N_1 \approx N_2$ and $\theta = c/N_2$. Then δ is small compared with N_2 as long as m_2 is $o(\sqrt{N_2})$. Hence the denominator in equation (9) is essentially constant (which can be made arbitrarily large by taking c to be large). However, if m_2 grows at all with N_2 then so does $\max_2$ and the value in equation (9) can become large. Hence, in practice, it is necessary that m_2 and $\max_2$ be (rather small) constants. The random walk therefore covers a distance $O(\sqrt{N_2})$ in the P_2 component.

Again, even making the over-cautious assumption that all steps in walks which contain bad points of type 2 are wasted, by choosing θ to be sufficiently large one can ensure that only a very small proportion of walks can have bad points of type 2 (at least, this is true when $N_1 \approx N_2$). Hence one may assume that, say, 1% of the walks are wasted (and hence the algorithm runs in about 1.01 times the theoretical prediction of the time).

We now tabulate the complexity statements we have obtained. We give the total number of group operations, but note that the algorithm can be easily parallelised giving linear speedup in the number of processors. The factor $1+\epsilon$ here is not the same as $1+o(1)$: As mentioned above, there can be a non-negligible proportion of bad points of types 1 and 2 (even asymptotically as N tends to infinity). There is also the fact that one never expects a pseudorandom walk of the type considered in this paper to have exactly the behaviour of a random walk (and so there is a small correction factor to include in the birthday paradox). This latter issue is discussed at length by Teske [20]; for example her Table 3 suggests that if $n_S = 16$ then the expected number of trials before finding a collision is 1.01 times more than that predicted by the birthday paradox for a random map. Hence, the actual values for ϵ in practice might be between 0.02 and 0.04.

Conjecture 1. The following table gives the expected total number of group operations for the original and improved Gaudry-Schost algorithms to solve the 2-dimensional DLP in the average and worst cases. Here $\epsilon > 0$ denotes a small constant (not necessarily the same value in all places).

Name of Algorithm	Average Case	Worst Case
Original Gaudry-Schost	$2.45(1+\epsilon)\sqrt{N} + \frac{1}{\theta}$	$3.58(1+\epsilon)\sqrt{N} + \frac{1}{\theta}$
Improved Gaudry-Schost	$2.38(1+\epsilon)\sqrt{N} + \frac{1}{\theta}$	$2.38(1+\epsilon)\sqrt{N} + \frac{1}{\theta}$

5 Higher Dimensions

It is clear that the algorithm can be used to solve the multidimensional DLP in any dimension. The main task is to choose a suitable pseudorandom walk. Indeed, there is an important issue that occurs with θ as d increases.

Recall that the Gaudry-Schost algorithm expects to store approximately $\theta(2^d/3^{d/2})\sqrt{\pi N}$ distinguished points. Hence, it is usual to assume that $\theta = c/\sqrt{N}$ for some large constant c. If using a pseudorandom walk which moves only forwards in some component then the walk will cover a distance $O(1/\theta) = O(\sqrt{N})$ steps. When $d \geq 3$, if $N_1 \approx N_2 \approx N_3$, then this distance is longer than the sides of the orthotope in question. Hence, walks would go outside the region in which the tame-wild birthday paradox is being applied and the complexity of the algorithm would deteriorate. Even when $d = 2$, as we have seen, it is necessary to ensure that θ is sufficiently large to have an algorithm which performs well.

As a result, when $d \geq 3$ it is necessary to use pseudorandom walks with 'side-to-side' steps (i.e., with both positive and negative steps and with average zero) in every component. If the mean absolute step size is constant then, by Lemma 3, the expected distance travelled in any component after $O(N^{1/2})$ steps is $O(N^{1/4})$, which is OK when $d = 3$ and $N_1 \approx N_2 \approx N_3$. When $d = 4$ and $N_1 \approx N_2 \approx N_3 \approx N_4$ then the walks are still OK, as long as θ is sufficiently large (i.e., as long as one can store sufficiently many distinguished points).

However for $d \geq 5$ the issue of walks stepping outside the region of interest re-appears and it cannot be resolved by using 'side-to-side' walks. Pollard [16] has suggested a solution to this problem which we describe in the appendix.

6 Applications

The 2-dimensional DLP arises in algorithms for computing the number of points on genus 2 curves over finite fields [12]. One uses a Schoof-type algorithm to get information about the coefficients of the characteristic polynomial of the Frobenius modulo some integer, and then uses a baby-step-giant-step algorithm to complete the calculation. Gaudry and Schost [9] developed their low-memory algorithm precisely for this application. These ideas have also be used by Weng [23]. Our results will therefore give an improvement to algorithms of this type.

This approach can be used to count points on curves of any genus (though for curves of sufficiently large genus one might also exploit subexponential algorithms). Depending on the amount of information obtained from the Schoof part of the algorithm, the remaining computation could be a d-dimensional DLP with $d \geq 3$. Our methods would also give an improvement here.

The multidimensional discrete logarithm problem also arises explicitly in the work of Brands [2] and in Section 4 of Cramer, Gennaro and Schoenmakers [5]. The latter paper notes that the problem can be solved using a baby-step-giant-step algorithm but does not mention the possibility of a low-memory or parallelisable algorithm.

The Gallant, Lambert and Vanstone (GLV) method [8] speeds up elliptic curve arithmetic by rewriting $[n]P$ as $[n_1]P + [n_2]\psi(P)$ for some endomorphism ψ and where $|n_1|, |n_2| \approx \sqrt{n}$. The integers n_1, n_2 are found by solving the closest vector problem in a lattice. An alternative is to choose "smaller" $n_1, n_2 \in \mathbb{Z}$ directly, rather than choosing a random integer n and rewriting it. Solving the DLP for points generated by the GLV method can be phrased as a multidimensional DLP. The methods of this paper imply that n_1 and n_2 cannot

be chosen to be too small. See Galbraith and Scott [7] and Galbraith, Lin and Scott [6] for examples of the GLV method with $d > 2$.

Another approach to efficient elliptic curve cryptography is to use Koblitz curves [11] (i.e., ordinary elliptic curves E over $\mathbb{F}_2$, considering the group $E(\mathbb{F}_{2^m})$). We rewrite $[n]P$ as

$$\sum_{i=0}^{L} n_i \tau^i(P)$$

where $n_i = \{-1, 0, 1\}$ and $\tau(P) = (x(P)^2, y(P)^2)$ is the 2-power Frobenius map. Since

$$\sum_{i=0}^{L} n_i \tau^i \equiv a + b\tau \text{ in } \mathbb{Z}[\tau]/(\tau^2 \pm \tau + 2)$$

where $|a| < 3\sqrt{2^L}$ and $|b| < 2\sqrt{2^L}$ as shown by Benits [1], solving the DLP can again be phrased as a multidimensional DLP i.e., $Q = [a]P + [b]\tau(P)$. It follows that L cannot be chosen to be too small. The same ideas can be applied on genus 2 curves over $\mathbb{F}_2$ leading to a 4-dimensional DLP.

7 Conclusion

We have presented an improvement to the algorithm given by Gaudry and Schost for solving the 2-dimensional DLP as well as extending the algorithm to the multidimensional DLP. We have also given further depth to the analysis given by Gaudry and Schost [9] specifically for the cases where $d > 2$. In Section 6 we have seen just a smattering of applications of this low-memory algorithm. An open problem is to investigate how best to exploit the algorithm when the search space is not an orthotope but a multidimensional 'arrowhead' which arises in the case of point counting on curves of genus 2 and higher.

Acknowledgements

We thank John Pollard, Pierrick Gaudry and the anonymous referees for helpful feedback and suggestions. We also thank our sponsors.

References

1. W. Benits Jr. *Applications of Frobenius expansions in elliptic curve cryptography.* PhD thesis, Royal Holloway, University of London, 2008.
2. S. Brands. An efficient off-line electronic cash system based on the representation problem, 1993. CWI Technical Report CS-R9323.
3. J. H. Cheon, J. Hong, and M. Kim. Speeding up the Pollard rho method on prime fields. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 471–488. Springer-Verlag, 2008.
4. E. G. Cofman, P. Flajolet, L. Flatto, and M. Hofri. The maximum of a random walk and its application to rectangle packing. Technical report, INRIA, 1997.
5. R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In W. Fumy, editor, *EUROCRYPT '97*, volume 1233 of *LNCS*, pages 103–118. Springer, 1997.
6. S. D. Galbraith, X. Lin, and M. Scott. Endomorphisms for faster elliptic curve cryptography on a large class of curves. In A. Joux, editor, *Eurocrypt 2009*, volume 5479 of *LNCS*, pages 518–535, Springer-Verlag, 2009.
7. S. D. Galbraith and M. Scott. Exponentiation in pairing-friendly groups using homomorphisms. In S. D. Galbraith and K. G. Paterson, editors, *Pairing 2008*, volume 5209 of *LNCS*, pages 211–224. Springer-Verlag, 2008.
8. R. Gallant, R. Lambert, and S. Vanstone. Improving the parallelized Pollard lambda search on binary anomalous curves. *Mathematics of Computation*, 69:1699–1705, 2000.
9. P. Gaudry and E. Schost. A low-memory parallel version of Matsuo, Chao and Tsujii's algorithm. In D. A. Buell, editor, *Proceedings of Algorithm Number Theory Symposium - ANTS VI*, volume 3076 of *LNCS*, pages 208–222. Springer-Verlag, 2004.

10. 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 A. J. van der Poorten and A. Stein editors, *Algorithmic Number Theory, 8th International Symposium, ANTS-VIII*, volume 5011 of LNCS, pages 402–415, Springer-Verlag, 2008.
11. N. Koblitz. CM-curves with good cryptographic properties. In J. Feigenbaum, editor, *CRYPTO '91*, volume 576 of *LNCS*, pages 279–287. Springer-Verlag, 1991.
12. K. Matsuo, J. Chao, and S. Tsujii. An improved baby step giant step algorithm for point counting of hyperelliptic curves over finite fields. In C. Fiecker and D. Kohel, editors, *Proceedings of Algorithm Number Theory Symposium - ANTS V*, volume 2369 of *LNCS*, pages 461–474. Springer-Verlag, 2002.
13. K. Nishimura and M. Sibuya. Probability to meet in the middle. *Journal of Cryptology*, 2:13–22, 1990.
14. J. M. Pollard. Monte Carlo methods for index computation mod p. *Mathematics of Computation*, 32(143):918–924, 1978.
15. J. M. Pollard. Kangaroos, Monopoly and discrete logarithms. *Journal of Cryptology*, 13:437–447, 2000.
16. J. M. Pollard. Remarks on discrete logs. Private Communication, August 2009.
17. R. S. Ruprai. Improvements to the Gaudry-Schost algorithm for multidimensional discrete logarithm problems and applications. PhD Thesis, Royal Holloway University of London, 2009.
18. D. Shanks. Class number, a theory of factorization and genera. In *Proc. Symposium in Pure Mathematics*, volume 20, pages 415–440, 1971.
19. D. Stinson. *Cryptography: Theory and practice*. Chapman & Hall/CRC, 3rd edition, 2006.
20. Edlyn Teske. On random walks for Pollard's rho method. *Mathematics of Computation*, 70(234):809–825, 2001.
21. P. C. van Oorschot and M. J. Wiener. Parallel collision search with application to hash functions and discrete logarithms. In *ACM Conference on Computer and Communications Security*, pages 210–218, 1994.
22. P. C. van Oorschot and M. J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12:1–28, 1999.
23. Annegret Weng. A low-memory algorithm for point counting on picard curves. *Designs, Codes and Cryptography*, 38(3):383–393, 2006.

A Pollard's Method for Large Dimensions

As mentioned in Section 5, when $d \geq 5$ walks of $O(\sqrt{N})$ steps will usually move outside the tame and wild regions. This is a major problem for the Gaudry-Schost algorithm. Pollard [16] has suggested a way to deal with this problem, and we briefly sketch it here.

We describe a parallel algorithm for the case $d = 1$. Clearly the algorithm can be serial or parallel, and applies to all d.

Let G be the group in question and let $N_1, \ldots, N_d$ be the interval sizes in the d-dimensional DLP with $N_1 \leq N_2 \leq \cdots \leq N_d$. Let $N = \prod_{i=1}^{d}(2N_i + 1)$ as usual. It is necessary to define two sets $D \subset S \subset G$, namely *distinguished points* and *special points*. As usual, the probability that a uniformly chosen $x \in G$ is distinguished should be $\theta = c/\sqrt{N}$ for some constant c. The probability p that a uniformly chosen $x \in G$ is special should be much higher, certainly $p > 1/N_1^2$. If this holds then one expects to find a special point after fewer than N_1^2 trials and a side-to-side random walk will cover distance $O(N_1)$ in that many steps and so still has a chance to be inside the region of interest.

We define a mapping $F : G \to G$ as follows. Let x be a point of G.

1. When x is a non-special point, as usual we have $F(x) = x + [r]P$ where r is small, and $[r]P$ is taken from a small table.
2. When x is a special point, we start a new walk from a new random point $[s]P$ or $Q + [s]P$ where s (and the choice tame/wild) are made determinstically and depending only on x (for example using a hash function). It is necessary that there are a large range of possible values for s.
   The computation of $[s]P$ can be done by exponentiation or by multiplying elements from $k \geq 3$ tables of size $N^{1/k}$. The second method does not have constant storage, but we can make the storage as small as we wish.
3. When x is a distinguished point we store x together with: a link to the last distinguished point on this processor, and the distance (number of steps) between the current and last points.

When a distinguished point is repeated, two sequences have met at some point y. If we find y, we have two representations of the same group element as $[a]P$ or $Q+[b]P$. With probability $1/2$, we have one of each type and can solve for the discrete logarithm. Otherwise we continue until a tame-wild collision is found.

We can easily find y by a small storage process. We know the two preceding distinguished points x_1 and x_2, on the two processors, and the distances travelled to the endpoint x. Suppose the distance from x_1 to x is longer than from x_2 to x. Advance the point x_1 to a point x_1' of the same distance to x as x_2. Now advance the points x_1' and x_2 together until they meet at y.

The final part of the algorithm requires an expected $1/\theta = O(\sqrt{N})$ steps and cannot be parallelised.