Noname manuscript No.
(will be inserted by the editor)

Recent progress on the elliptic curve discrete logarithm problem

Steven D. Galbraith · Pierrick Gaudry

Received: date / Accepted: date

Abstract We survey recent work on the elliptic curve discrete logarithm problem. In particular we review index calculus algorithms using summation polynomials, and claims about their complexity.

Keywords Elliptic curve discrete logarithm problem (ECDLP) · Summation polynomials · Pollard rho · Index calculus

Mathematics Subject Classification (2000) 11Y16 · 11G20 · 14G15 · 13P10 · 14G50 · 11T71 · 14H52

1 Introduction

Let E be an elliptic curve over a finite field $\mathbb{F}_q$, where $q = p^n$ and p is prime. The *elliptic curve discrete logarithm problem (ECDLP)* is the following computational problem: Given points $P, Q \in E(\mathbb{F}_q)$ to find an integer a, if it exists, such that $Q = aP$. This problem is the fundamental building block for elliptic curve cryptography and pairing-based cryptography, and has been a major area of research in computational number theory and cryptography for several decades.

There are many excellent books that provide a detailed background to elliptic curve cryptography, for example [3, 12, 13, 42, 58, 114]. Hence, the goal of this article is to survey developments from within the last five years or so. In particular we wish to highlight some open questions and areas where more work is needed. We assume the reader already has a good knowledge of elliptic curves and algorithms. We focus on the case of elliptic curves, but occasionally this involves mention of higher genus curves and their divisor class groups. However, we do not attempt to discuss all recent work regarding the DLP for curves of genus greater than one.

S. D. Galbraith
University of Auckland, New Zealand.
E-mail: s.galbraith@math.auckland.ac.nz

P. Gaudry
CNRS, Université de Lorraine and Inria, Nancy, France.
E-mail: pierrick.gaudry@loria.fr

There has been a substantial amount of recent work on efficient implementation of elliptic curves, but we do not present any of it in this survey. We also do not discuss pairing-based cryptography. Instead, we focus entirely on the ECDLP and related computational problems.

An active area of research is index calculus algorithms based on summation polynomials. Recently several papers have suggested potential subexponential algorithms for the ECDLP. However, we believe that elliptic curves over characteristic two fields $\mathbb{F}_{2^n}$ of prime degree n are not threatened by such methods and are still safe for use.

The paper is organised as follows. Sections 2 and 3 recall basic facts on elliptic curves and computational problems. Sections 4 and 5 discuss generic algorithms for the ECDLP. Section 6 sketches the principle of index calculus algorithms for the discrete logarithm problem. Sections 7 and 8 introduce the ideas of Weil descent and summation polynomials. Section 9 discusses point decomposition in greater detail, while Section 10 reports on open questions.

2 Basic notation on elliptic curves

An elliptic curve is given in *Weierstrass model* as

$$E : y^2 + A_1xy + A_3y = x^3 + A_2x^2 + A_4x + A_6$$

such that the discriminant is non-zero. If $P = (x, y)$ is a point on such a curve then $-P = (x, -y - A_1x - A_3)$. The point at infinity is denoted ∞. Other curve models such as Montgomery and twisted Edwards may be useful for efficient implementation, but from the point of view of the ECDLP the various models do not play such a prominent role as one can usually switch between them as required.

For future reference we recall the *binary Edwards model* [9], with $D_1, D_2 \in \mathbb{F}_{2^n}$ such that $D_1 \neq 0$ and $D_2 \neq D_1^2 + D_1$,

$$E : D_1(x + y) + D_2(x^2 + y^2) = xy + xy(x + y) + x^2y^2.$$

This curve has identity $(0, 0)$. For $P = (x, y)$ we have $-P = (y, x)$. There is a point $T = (1, 1)$ on E of order 2, and $(x, y) + T = (x + 1, y + 1)$.

For a point P on any elliptic curve and a non-negative integer a we define aP to be $P + P + \cdots + P$ (a times). We extend this definition to all $a \in \mathbb{Z}$ using $aP = (-a)(-P)$. The ECDLP is: Given P, Q on an elliptic curve to find an integer a, if it exists, such that $Q = aP$.

A *subfield curve* is an elliptic curve whose coefficients lie in $\mathbb{F}_q$ but where the DLP instance is in the larger group $E(\mathbb{F}_{q^n})$ where $n > 1$. Note that $\#E(\mathbb{F}_q)$ divides $\#E(\mathbb{F}_{q^n})$ and, to ensure that $\#E(\mathbb{F}_{2^n})$ has a large prime factor, n is usually chosen to be prime. The q-power Frobenius map $\pi_q(x, y) = (x^q, y^q)$ acts on $E(\mathbb{F}_{q^n})$.

The mathematical theory of elliptic curves is extremely rich, and there are several mathematical concepts that have been used to give results on the ECDLP.

- For the so-called anomalous curves (elliptic curves $E(\mathbb{F}_p)$ where p is prime such that $\#E(\mathbb{F}_p) = p$) the p-adic logarithm map allows to solve the ECDLP very easily. This approach can also be formulated in terms of a space of differentials. The details of the algorithm are by now standard and can be found in [3, 42]. Since such curves are an extremely special case, we do not discuss these methods further.

- When E is not anomalous the Weil and Tate-Lichtenbaum pairings transform the ECDLP in $E(\mathbb{F}_q)$ into an instance of the discrete logarithm problem in the multiplicative group of a finite field $\mathbb{F}_{q^k}$. For certain special elliptic curves the finite field $\mathbb{F}_{q^k}$ is small enough that the resulting instance can be solved using an index calculus algorithm. Again, the details are standard and the only major recent developments have been the impressive progress on finite field discrete logarithms.
- The Xedni calculus algorithm is a variant of index calculus proposed by Silverman that is based on lifting elliptic curves from finite fields to number fields. The paper [72] analysed the algorithm and explained why this approach is not likely to be successful. We do not discuss it further in this article.
- By an algorithm of Galbraith [41], further improved in [44], it is generally possible to efficiently map a discrete logarithm problem in an elliptic curve E to another E' in the same isogeny class where the problem might be easier to solve (also see [73]).
- Frey [39, 40] has discussed how the discrete logarithm problem in finite fields can be expressed in terms of the Brauer group and "invariants" of local Brauer groups (also see [98]). Huang and Raskind [65] have expressed the discrete logarithm problem (in finite fields and for elliptic curves over finite fields) in terms of the "signature calculus". A talk by Karl Rubin (unpublished) at a conference in Ascona in May 2014 presented joint work with Alice Silverberg on using Kolyvagin systems to transfer the ECDLP to the DLP in certain finite fields.

  All these approaches to the ECDLP reveal deep connections within number theory, but at the moment none of them has been demonstrated to have any practical impact, so we do not discuss them here.
- One can construct algebraic-geometry codes based on elliptic curves such that computing the minimum distance of the code is equivalent to solving the ECDLP. Driencourt and Michon [32] first noticed this connection, and it was rediscovered by Cheng [21]. Augot and Morain [2] have explored the use of these ideas to solve the ECDLP, but so far the approach does not appear to be very promising.

3 Computational Problems

As mentioned, our focus in this paper is the ECDLP: Given $P, Q \in E(\mathbb{F}_q)$ to find an integer a, if it exists, such that $Q = aP$. However, there are many other computational problems of interest in elliptic curve cryptography. The following two are the most important. The *Computational Diffie-Hellman problem* (CDH) is: Given $P, aP, bP \in E(\mathbb{F}_q)$ to compute abP. The *Decisional Diffie-Hellman problem* (DDH) is: Given $P, aP, bP, Q \in E(\mathbb{F}_q)$ to determine if $Q = abP$. Currently the only known way to solve the Computational Diffie-Hellman problem is to solve the ECDLP. The Decisional Diffie-Hellman problem can be solved using pairings for some special elliptic curves, but in the general case the only algorithm known to solve it requires solving the ECDLP. Hence, in practice, the study of algorithms for the ECDLP is the main way to assess the security of cryptographic applications of elliptic curves.

Another problem that arises in certain cryptographic protocols is the *discrete logarithm problem in an interval*: Given points $P, Q \in E(\mathbb{F}_q)$ and an integer $N < \#E(\mathbb{F}_q)$ to find a if it exists such that $Q = aP$ and $0 \leq a < N$.

In practice, many users will choose their public keys with respect to a fixed group $E(\mathbb{F}_q)$ and generator P. This leads to the study of the *multiple ECDLP*: Given $\{Q_1, \ldots, Q_L\} \subseteq E(\mathbb{F}_q)$ to compute $a_1, \ldots, a_L$ such that $Q_i = a_i P$ for all $1 \leq i \leq L$. Clearly this can be done in at most L times the cost of solving a single instance, but can one do better? We will discuss this problem in the coming sections. One can also consider the problem of solving just one of the L instances: Can one do this significantly faster than just taking the first instance Q_1 and solving it? Kuhn and Struik [89] do consider the "one out of L" problem, but we are not aware of any algorithm that solves that problem in less time than a single ECDLP instance.

A variant of the Diffie-Hellman problem introduced by Boneh and Boyen [14] is to compute a when given $P, aP, a^2P, a^3P, \dots, a^dP$. Problems of this type (including the simpler case of being given P, aP, a^dP) are sometimes called "discrete logarithm problems with auxiliary inputs". Cheon [22, 23] has given an improved algorithm to compute the ECDLP in this case (we sketch the ideas at the end of Section 4).

Finally one can consider "interactive" computational problems, which means that an algorithm to solve a problem is allowed to make some queries to an oracle. One example of such an oracle that arises in some applications is a static-Diffie-Hellman oracle: this is an oracle O that on input $Q \in E(\mathbb{F}_q)$ computes aQ for a fixed (static) integer a. One can consider the problem of computing a given P and such an oracle O. This problem was considered by Brown and Gallant [18]. One can also consider "one-more" problems, which are of the following flavour: Given $P, a_1P, \dots, a_LP$ and an oracle that solves the ECDLP, to compute $a_1, \dots, a_L$ using at most $L-1$ queries to the oracle. Such problems are surveyed in several papers by Koblitz and Menezes [83, 84], also see Joux, Lercier, Naccache and Thomé [75] and Granger [55]. We do not discuss such problems further in this paper, except briefly in Section 9.3.

4 Baby-step-giant-step

The main case of interest is elliptic curves over finite fields $\mathbb{F}_q$ whose group order is divisible by a prime $r > 2\sqrt{q}$, and where the points $P, Q \in E(\mathbb{F}_q)$ both have order r. It follows that there is a solution $a \in \mathbb{Z}/r\mathbb{Z}$ to $Q = aP$.

The baby-step-giant-step algorithm is based on the observation that, taking $M = \lceil \sqrt{r} \rceil$, one can write $a = a_0 + a_1M$ where $0 \le a_0 < M$ and $0 \le a_1 < M$. The idea is to compute and store all points a_0P (these are the "baby steps") and then to sequentially compute $Q - a_1(MP)$ (the "giant steps") and seek a match in the list. The point MP is computed once at the start of the algorithm. The algorithm performs $O(\sqrt{r})$ group operations and requires $O(\sqrt{r})$ group elements of storage. The asymptotic complexity $O(\sqrt{r})$ group operations of this algorithm is optimal due to Shoup's lower bound [109] (building on work of Babai-Szemerédi [4] and Nechaev [97]) on the complexity of a "generic algorithm" for solving the DLP.

Pollard (Section 3 of [101]) proposed to "interleave" the computations to improve the average-case performance. This gives an algorithm with average-case running time of $\frac{4}{3}\sqrt{r}$ group operations.

For elliptic curve groups in Weierstrass model one can exploit the fact that if $P = (x, y) \in E(\mathbb{F}_q)$ then $-P$ is easily computed and has the same x-coordinate. Hence one can improve Pollard's method. One takes $M = \lceil \sqrt{2r} \rceil$ so that $a = a_0 + a_1M$ where $-M/2 \le a_0 \le M/2$ and $0 \le a_1 < r/M \approx \sqrt{r/2} \approx M/2$. One then computes two lists $\{x(a_0P) : a_0 = 0, 1, 2, \dots\}$ and $\{x(Q - a_1(MP)) : a_1 = 0, 1, 2, \dots\}$ until there is a match on x-coordinates (meaning that $Q - a_1(MP) = \pm a_0P$). These lists contain one element for each equivalence class $\{P, -P\}$ of points. Using Pollard's analysis, the average-case running time is $\frac{4}{3}\sqrt{r/2} \approx 0.943\sqrt{r}$ group operations.

A natural question is to give a tight lower bound on the number of group operations for any algorithm of this type. The original baby-step-giant-step algorithm is non-optimal, since the DLP can only be solved when there is a match between an element in the list of baby steps and an element in the list of giant steps. So if a total of k elements have been computed (two lists of size $k/2$) then at most $(k/2)^2$ possible instances of the DLPs would be solved. Chateauneuf, Ling and Stinson [20] considered a "best possible" baby-step-giant-step algorithm under an unrealistic model where computing an arbitrary element $aP+bQ$ is counted as a single operation. The idea is to compute one list of points $\{a_iP+b_iQ\}$ such that every collision $a_iP+b_iQ = a_jP+b_jQ$ corresponds to a different value for the discrete logarithm. Hence, if the list has size k then one can solve $\binom{k}{2} \approx k^2/2$ instances of the DLP; twice as many as the basic baby-step-giant-step algorithm. The paper [20] is mainly about the combinatorics of choosing suitable pairs (a_i, b_i). It is a straightforward exercise to verify that, assuming one can construct such lists and under this unrealistic computational model, the worst-case complexity of the DLP in a group of size r is $(\sqrt{2} + o(1))\sqrt{r}$ operations, and the average-case complexity is $\frac{2}{3}(\sqrt{2} + o(1))\sqrt{r} \approx (0.943 + o(1))\sqrt{r}$ operations. For elliptic curves, when exploiting inversion, the natural lower bound for the average-case would become $(\frac{2}{3} + o(1))\sqrt{r}$ operations. Both these lower bounds are better than what could be expected for any method (such as Pollard rho, see below) that relies on the birthday paradox.

As a step towards such a "best possible" algorithm, Bernstein and Lange [8] have given an interesting variant of the baby-step-giant-step algorithm. They consider three lists $\{a_0P : a_0 = 0, 1, 2, \dots\}$, $\{Q+a_1(MP) : a_1 = 0, 1, 2, \dots\}$ and $\{2Q-a_2((M+1)P) : a_2 = 0, 1, 2, \dots\}$, where $M \approx \sqrt{r}/2$. A match between any pair of lists can lead to a solution to the DLP, so once k group elements have been computed we can hope to solve around $3(k/3)^2 = k^2/3$ instances of the DLP. Since $k^2/4 < k^2/3 < k^2/2$ we expect the method to be better than the basic algorithm but not to match the "ideal" lower bound. The method is described as "two grumpy giants and a baby", since the latter two walks are "giant step" walks but in opposite directions. Some theoretical analysis is given in [8], but no precise statement is given of the average-case asymptotic performance. The theoretical arguments and experimental results suggest the algorithm has slightly better average-case running time than Pollard rho, perhaps around $(1.2 + o(1))\sqrt{r}$ group operations. The paper [48] analyses the algorithm in the case of elliptic curves and exploiting inversion, and also considers efficient ways to perform baby-step-giant-step algorithms for elliptic curves by computing "blocks" of points and sharing inversions using the Montgomery trick. It is an open question to determine the exact average-case running time of the "grumpy giants" algorithm and to develop algorithms whose running time is closer to the theoretical lower bound.

We now sketch the technique due to Cheon [22, 23] and Brown-Gallant [18] for solving ECDLP instances $P, Q_1 = aP, Q_d = a^dP$ where P has order r and $d \mid (r-1)$. Fix $z \in (\mathbb{Z}/r\mathbb{Z})^*$ of order equal to $(r-1)$, so that z^d has order $(r-1)/d$. Since a^d has order modulo r dividing $(r-1)/d$, we have $a^d \equiv (z^d)^x \pmod{r}$ for some integer $0 \le x < (r-1)/d$. Writing $M = \lceil \sqrt{(r-1)/d} \rceil$ and $x = x_0 + Mx_1$ with $0 \le x_0, x_1 < M$ we have $a^dP = (z^d)^{x_0}(z^{Md})^{x_1}P$. Hence one can compute a list of values $z^{-dx_0}Q_d$ and a list of values $(z^{dM})^{x_1}P$ and find in $O(\sqrt{r/d})$ steps the matching pair (x_0, x_1). Writing $x = x_0 + Mx_1$ we have $a^d \equiv z^{dx} \pmod{r}$. To find a we write $a = z^k$ and note that $k = x + y(r-1)/d$ for some $0 \le y < d$. By a similar method based on $Q_1 = aP$ one computes y in $O(\sqrt{d})$ steps and hence computes a. Overall we compute

a in $O(\max\{\sqrt{r/d}, \sqrt{d}\})$ group operations. In the extreme case where there is a factor $d \mid (r-1)$ with $d \approx r^{1/2}$ and one is given aP and $a^d P$ then one can solve the ECDLP in $O(p^{1/4})$ steps. Cheon also presents a variant for the case when $d \mid (p+1)$. For generalisations of this method see [24, 81, 88, 105, 103, 82].

5 Pollard rho and kangaroo

The baby-step-giant-step algorithm requires large storage and is hard to parallelise or distribute over the internet. The rho and kangaroo algorithms require less storage and can be distributed. The basic idea is to reduce the discrete logarithm problem to the problem of collision-finding, and then use low-storage collision-detection methods. In the rho algorithm one seeks a collision of the form $aP + bQ = a'P + b'Q$ while in the kangaroo algorithm one seeks a collision of the form $aP = Q + a'P$. Both algorithms exploit pseudorandom walks. We do not present all the details of these algorithms as there are several good references [3, 42], but the main principle is to design stateless pseudorandom walks in the group so that:

1. The next group element in the walk is computed as a deterministic function of the current group element;
2. The cost of each step in the walk is approximately the cost of a single group operation.

The heuristic analysis of the Pollard rho algorithm is based on the birthday paradox. Van Oorschot and Wiener [99] explained how to use "distinguished points" to get heuristic average case expected running time (in the serial case) of $(\sqrt{\pi/2} + o(1))\sqrt{r}$ group operations. It is natural to believe that this running time is "optimal", in the sense that no algorithm based on finding collisions in a set of size r should be able to do better than the birthday paradox. Note that the worst-case running time for Pollard rho is unbounded.

When doing experiments one immediately notices that the variance in the running time of the Pollard rho and kangaroo (see Section 5.1) algorithms is rather large. Hence, the precise expected running time is only a useful guideline when one is performing many ECDLP computations. However, if one is solving L ECDLP instances in the same group one can do better than performing L independent instances of Pollard rho. Kuhn and Struik [89] studied re-using all previous distinguished point values when solving many instances and showed that if $L < r^{1/4}$ then one can solve all L instances in approximately $\sqrt{2rL}$ group operations. The instances are solved consecutively, but note that, since all distinguished points are stored, the storage cost is greater than performing L computations in serial. Further discussion on this problem was given by Hitchcock, Montague, Carter and Dawson [61], Bernstein and Lange [6, 7] and Fouque, Joux and Mavromati [38]. In particular, [6, 7] (also see [69, 63]) study the case where precomputation costs are ignored.[1] They show that one can heuristically achieve running time of $O(r^{1/3})$ group operations for each instance of the ECDLP after an $O(r^{2/3})$ precomputation, and with a program size of $O(r^{1/3}\log(r))$ bits. The algorithms in [6, 7] solve the ECDLP instances sequentially whereas the approach in [38] collects relations among the ECDLP instances until all (or almost all) instances

[1] This is sometimes called the "non-uniform" model, but we do not discuss such interpretations in this paper. Note that an algorithm that stores a table of all discrete logs does not fit the model since the program length is $O(r\log(r))$ bits.

Authors	Bit size	Year	Field/Method	Hardware
Monico [19]	108	2002	Large p/no orbits	CPU
Monico [19]	108	2004	Char 2/no orbits	CPU
Bos et al. [16]	111	2009	Large p/no orbits	PS3
Wenger, Wolfger [115]	112	2014	Koblitz/orbits	FPGA
Wenger, Wolfger [116]	113	2015	Char 2/orbits	FPGA

Table 1 Summary of record ECDLP computations.

are simultaneously solved together. Kuhn and Struik conjectured an $\Omega(\sqrt{rL})$ lower bound on the complexity of solving L instances of the ECDLP in the same group, and this was recently proved in the generic group model by Yun [118].

Gallant, Lambert and Vanstone [49] and Wiener and Zuccherato [117] showed how to exploit automorphisms of the group to get a faster algorithm. Essentially the idea is to take an automorphism (for example, the map $\psi(P) = -P$ on an elliptic curve, or a Frobenius map) and consider the rho algorithm on the set of orbits under ψ. Note that the set of orbits is not a group, but one can still exploit the group operation to construct pseudorandom walks. There are two ways to achieve this. One way is to specify a unique representative of each orbit, and to define the next step in the walk based on this representative. The other way is to define walks using a "next step function" $f : E \to E$ that is well-defined on orbits (i.e., $\{\psi^i(f(P))\} = \{f(\psi^i(P))\}$). In both cases the algorithm should be sped-up by approximately $\sqrt{l}$ where l is the average size of orbits. These approaches are slightly different to how inverses are used to speed up the baby-step-giant-step algorithm: there we did not need to construct a "well-defined walk" but only detect a match between two lists up to sign.

One difficulty that arises when working on orbits is useless cycles. For example, the steps in a walk might be: $P \mapsto P + R \mapsto -(P + R) \mapsto -(P + R) + R = -P \mapsto P$, where the symbol $\mapsto$ denotes either a step in the walk (by adding a point R) or the change of sign from moving to the unique representative of the orbit. Bos, Kleinjung and Lenstra [17] raised some practical issues that suggest the additional overhead of dealing with useless cycles might result in the speedup being less than $\sqrt{l}$. However, Bernstein, Lange and Schwabe [10] showed how to organise the computation to minimise such concerns. For further discussion see [15].

Most computations use pseudorandom walks where each step is an addition. But walks that include doubling operations can be useful in practice, and more resilient to short cycles. Zhang and Wang [119] suggested to replace doubling steps with more efficient point halving steps when working with elliptic curves in characteristic two.

The Pollard rho algorithm is the method used to solve all large-scale ECDLP computations. We gather the recent results in Table 1. The phrase "Koblitz" denotes a curve E over $\mathbb{F}_2$ with group $E(\mathbb{F}_{2^n})$ where n is prime; in this case the Frobenius automorphism $\psi(x, y) = (x^2, y^2)$ acts on the group and so one gets orbits of size $2n$ from Frobenius and inversion. We write "orbits" if the algorithm exploited orbits under some automorphism (including $P \mapsto -P$ only) and "no orbits" if not.

In November 2009 an attempt was initiated [5] to solve the Certicom challenge ECC2K-130. This is an ECDLP challenge in $E(\mathbb{F}_{2^{131}})$ where E is defined over $\mathbb{F}_2$ and the group order is $4r$ for some 130-bit prime r. At time of writing (August 2015) the computation is still running.

It is worth noting that most papers on the Pollard rho and kangaroo algorithms rely on heuristic assumptions. One issue that has received a lot of attention is the effect on the running time due to the number of partitions used to define the walk: A first heuristic was proposed by Brent and Pollard in the context of integer factorisation; Blackburn and Murphy [11] rediscovered this idea in the case of the rho algorithm for the ECDLP; Section 3 of Bernstein and Lange [8] discusses a refinement of the idea for rho; Kijima and Montenegro [79] give a derivation of it and prove rigorous results for both the rho and kangaroo algorithms (and more). Experimental results confirm the analysis in those papers, but it is a challenging and interesting task to minimise the use of heuristics but still get good results about these algorithms. Ravi Montenegro and co-authors have had a number of further successes in this regard see [80, 94, 71].

5.1 Kangaroo and Gaudry-Schost methods

The kangaroo method is subtly different to the rho method. It is most suitable for the discrete logarithm problem in an interval of length N (where N is less than the order of the point P). For the kangaroo method, steps in the pseudorandom walk are relatively small jumps (in the rho method they are random jumps), and the algorithm is not analysed using the birthday paradox. Van Oorschot and Wiener [99] showed that the kangaroo method performs on average $(2 + o(1))\sqrt{N}$ group operations.

A third class of algorithm is due to Gaudry and Schost [52]. This algorithm uses "small jumps" and yet is analysed using the birthday paradox, so in some sense it "interpolates" the kangaroo and rho algorithms.

Using these algorithms, Galbraith, Pollard and Ruprai [45] improved the average case running time of the DLP in an interval to $(1.66 + o(1))\sqrt{N}$ group operations. Galbraith and Ruprai [46] also studied how to speed-up these methods when exploiting orbits under the inversion map $P \mapsto -P$.

An open question is to find a low-storage algorithm to solve the *low Hamming weight DLP*: Given points P, Q and integers m, w find a, if it exists, such that $Q = aP$, $0 \leq a < 2^m$, and the Hamming weight of the binary representation of a is at most w. The number of possible values for a is roughly $\binom{m}{w}$ so one would like an algorithm that performs roughly $\sqrt{\binom{m}{w}}$ group operations. No low-storage algorithm with that time complexity is known, whereas it is easy to find a baby-step-giant-step algorithm for this problem. For further discussion and references see Sections 13.6 and 14.8.1 of [42] and also the recent paper [91] which discusses this problem in groups of composite order.

6 Index calculus

6.1 General setting

The idea of index calculus algorithms is to reduce the discrete logarithm problem to linear algebra. Let G be a cyclic group of order r, that we denote additively to conform to the elliptic curve notations, and let P be a generator and Q be another element for which we want to compute the discrete logarithm. The simplest version of index calculus is as follows:

1. Define a subset $\mathcal{F}$ of G, called the factor base.
2. Collect relations:

   (a) Pick random integers a and b and compute $R = aP + bQ$;
   (b) Try to decompose R as a sum of elements of $\mathcal{F}$;
   (c) In case of success, $aP + bQ = \sum_{P_i \in \mathcal{F}} e_i P_i$, call it a relation, and store integers (a, b) and the vector (e_i) as a row of a matrix (the relation matrix);
   (d) Repeat the procedure until we have at least $\#\mathcal{F}$ relations.
3. Via linear algebra modulo r, compute a linear combination of the relations such that the right-hand-side vanishes; this leads to an equation $\lambda P + \mu Q = 0$ in G.
4. If μ is invertible modulo r, the discrete logarithm of Q is $\lambda/\mu \bmod r$.

For a given group G, the difficulty is to choose a factor base $\mathcal{F}$ that has the following properties, where the key difficulty resides in the decomposition step that must be fast and have a high success probability:

- The set $\mathcal{F}$ should not be too large, since we need to collect $\#\mathcal{F}$ relations.
- It should be the case that a large proportion of group elements can be written as a sum of elements in $\mathcal{F}$; otherwise Step 2b will fail too often.
- Given an arbitrary group element it should be efficient to decompose it as a sum in $\mathcal{F}$, or else decide that such a decomposition does not exist; otherwise each execution of Step 2b will take too long.

In general, the decomposition of an element will involve only a small number of factor base elements. Therefore the matrix is usually quite sparse, and appropriate linear algebra algorithms must be used (see Chapter 3.4 of Joux [74]).

The archetype of this algorithm is for the group $\mathbb{F}_p^*$ where p is a large prime. In that case, of course it would be easier to use a multiplicative notation for the group law. One sets $\mathcal{F} = \{p_1, \ldots, p_k\}$ to be the set of the first k primes. One can consider any group element $R \in \mathbb{F}_p^*$ as an integer in the range $1 \leq R < p$ and try to factor it as a product of primes. Denoting $L_p(a)$ a subexponential function $\exp(c \log(p)^a \log(\log(p))^{1-a})$ for some constant c, one takes $k = L_p(1/2)$. The set $\mathcal{F}$ has subexponential size and the probability that a random integer less than p can be written as a product of primes in $\mathcal{F}$ is $1/L_p(1/2)$. One therefore gets an algorithm with subexponential running time.

6.2 Two approaches to relations

In the literature describing index calculus algorithms for discrete logarithm computations, there are two strategies (assuming relations are converted into rows of a matrix):

- Make the relations depend on the target element whose discrete logarithm is sought, and solve a left-kernel linear algebra problem. This is how we have presented the method above (a vanishing linear combination of rows is a left-kernel element);
- Use relations that do not depend on the target element; solve a right-kernel linear algebra problem; and add an additional step to deduce the discrete logarithm of the target element. In Step 2, we would then define $R = aP$ for a random a, and replace Step 3 by a right-kernel computation, giving the discrete logarithms of all the factor base elements.

The first version is much easier to analyze rigorously. Indeed, provided that we have more relations than elements in the factor base, the left-kernel is non-trivial, and it is usually not too difficult to randomize the algorithm in such a way that we can prove that any non-zero kernel vector gives the discrete logarithm with a high probability.

The main drawback is that the (costly) linear algebra step is to be done for each discrete logarithm computation, even if several take place in the same group. In practice, especially when solving multiple instances of the ECDLP, the second version is preferred. Obtaining the discrete logarithms of all the factor base elements with a right-kernel computation is guaranteed only if the matrix has maximum possible rank (namely one minus the number of factor base elements). This rank-condition is rarely verified if we have just as many relations as the number of unknowns, so a heuristic approach is to run the linear algebra step only when many more relations than unknowns have been constructed. On the theoretical side, Pomerance [102] was one of the first to develop techniques to produce relations in a way that the rank condition is automatically fulfilled.

This second version requires an additional step which is often simple (in particular for the basic index calculus method sketched above when applied to $\mathbb{F}_p^*$). However, in some cases, and most notably for the number field sieve algorithm, finding the logarithm of a target element once the logarithms of the factor base elements are known is a non-trivial step; it is often called the "individual logarithm" stage or the "descent" (not to be confused with Weil descent).

6.3 An index calculus for elliptic curves?

The main difficulty when trying to apply the general index calculus technique to elliptic curves is to find a factor base $\mathcal{F}$ so that a large proportion of elements can be written as a sum of elements of $\mathcal{F}$, while having an efficient decomposition algorithm. In general, with basic combinatorics arguments using only the size of $\mathcal{F}$ and the number of elements allowed in the decomposition, the proportion of decomposable elements is rather easy to estimate (at least, heuristically). Having a fast algorithm for the decomposition is the place where it is always necessary to use the structure of $\mathcal{F}$ (and necessarily step away from generic algorithms). In the case of $\mathbb{F}_p^*$, there is a natural notion of "small element" due to the representation as an integer; and the decomposition algorithm is integer factorisation (e.g., using trial division followed by the Elliptic Curve method).

In the case of elliptic curves defined over prime fields, Semaev [107] proposed to use for $\mathcal{F}$ the set of points whose abscissa is small when viewed as an integer. However, the elliptic group law is deeply incompatible with the multiplication of the integers representing the abscissae of the points. As a consequence, right now, there is no known efficient decomposition algorithm for that choice of $\mathcal{F}$, and elliptic curves over prime fields remain unaffected by index calculus algorithms. This is no longer the case for some elliptic curves over extension fields, and this is the subject of the rest of this survey.

Finally, we mention the important fact that the index calculus idea can be applied efficiently to divisor class groups of smooth projective curves of genus $g > 1$ [59, 27]. For hyperelliptic curves one takes for $\mathcal{F}$ the set of reduced divisors whose first polynomial in the Mumford representation is an irreducible polynomial of small degree; there is an efficient decomposition algorithm based on factorization of polynomials and we get a subexponential discrete logarithm algorithm when the genus goes to infinity [1]. Even when the genus is as low as 3, this approach is faster than generic algorithms like Pollard rho [111, 53, 26, 30].

7 Weil descent

All recent progress on non-generic attacks on elliptic curves is related to the Weil restriction of scalars. The basic idea is quite simple: a polynomial equation defined over an extension field $\mathbb{F}_{q^n}$ can be re-written as n polynomial equations defined over $\mathbb{F}_q$.

Lemma 1 *Let q be a prime power, $n \geq 1$, and fix a vector space basis $\{\theta_1, \ldots, \theta_n\}$ for $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. Let $f(x_1, \ldots, x_m) \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]$. Let $R = \mathbb{F}_q[y_{i,j} : 1 \leq i \leq m, 1 \leq j \leq n]$. Then there exist unique polynomials $f_k(y_{i,j}) \in R$ for $1 \leq k \leq n$ such that*

$$f(y_{1,1}\theta_1 + \cdots + y_{1,n}\theta_n, \ldots, y_{m,1}\theta_1 + \cdots + y_{m,n}\theta_n) \ = \ \sum_{k=1}^{n} \theta_k f_k(y_{i,j}).$$

Furthermore, if $f(a_1, \ldots, a_m) = 0$ for some $a_1, \ldots, a_m \in \mathbb{F}_{q^n}$ then there exist $b_{i,j} \in \mathbb{F}_q$ such that $a_i = \sum_{j=1}^{n} b_{i,j}\theta_j$ and $f_k(b_{i,j}) = 0$ for all $1 \leq k \leq n$.

The lemma can be directly applied to the equation of an elliptic curve over $\mathbb{F}_{q^n}$, thus getting n equations in $2n$ indeterminates over $\mathbb{F}_q$, that is an n-dimensional algebraic variety. Since the elliptic curve group law can be transported to this variety, it is an (affine model of an) Abelian variety. The GHS attack [47, 51] relies on the idea of finding an algebraic curve $\mathcal{C}$ of genus $g \geq n$ as small as possible such that the Jacobian of $\mathcal{C}$ contains the target Abelian variety. If this is the case, the original elliptic curve discrete logarithm problem can be transferred into this Jacobian defined over $\mathbb{F}_q$, where the problem could become easier due to the subexponential index calculus. This has been shown to be successful for a few families of elliptic curves (see [92, 25, 31, 60, 112, 93] and Hess's survey in [13, Chapter VIII]). In general this approach fails, because the curves $\mathcal{C}$ that are constructed have a genus that is exponential in n instead of linear in n. There has not been a lot of progress in that area in recent years, as it becomes harder and harder to find new weak families. A recent preprint [70] did a rather systematic study showing that there is not so much to expect anymore.

Interestingly, the idea of the Weil restriction can also be used inside the decomposition problem of an index-calculus approach: using the formulae of the group law, decomposing a point R into $P_1 + \cdots + P_k$, can be written as a system of polynomial equations over $\mathbb{F}_{q^n}$, that can be converted into equations over $\mathbb{F}_q$ if $\mathcal{F}$ has a well-chosen algebraic description over $\mathbb{F}_q$. In the next section we introduce Semaev's summation polynomials, as they provide a way to compute this in practice.

8 Summation polynomials

Semaev's unpublished note [107] has been enormously influential on the field. Let E be an elliptic curve in Weierstrass model over a field $\Bbbk$ of odd characteristic. Let $m \in \mathbb{N}$. The m-th summation polynomial $\mathrm{S}_m(x_1, x_2, \ldots, x_m) \in \Bbbk[x_1, x_2, \ldots, x_m]$ has the following defining property: Let $X_1, X_2, \ldots, X_m \in \overline{\Bbbk}$. Then $\mathrm{S}_m(X_1, X_2, \ldots, X_m) = 0$ if and only if there exist $Y_1, Y_2, \ldots, Y_m \in \overline{\Bbbk}$ such that $(X_i, Y_i) \in E(\overline{\Bbbk})$ for all $1 \leq i \leq m$ and $(X_1, Y_1) + (X_2, Y_2) + \cdots + (X_m, Y_m) = 0$ on the curve.

Lemma 2 *(Semaev [107]) Let $E : y^2 = x^3 + A_4x + A_6$ be an elliptic curve in short Weierstrass model over a field of odd characteristic. The summation polynomials for E are given as follows.*

$$\begin{aligned} \mathrm{S}_2(x_1, x_2) &= x_1 - x_2 \\ \mathrm{S}_3(x_1, x_2, x_3) &= (x_1 - x_2)^2 x_3^2 - 2((x_1 + x_2)(x_1x_2 + A_4) + 2A_6)x_3 \\ &\quad +((x_1x_2 - A_4)^2 - 4A_6(x_1x_2)). \end{aligned}$$

For $m \geq 4$ let j be such that $1 \leq j \leq m - 3$, then

$$\mathrm{S}_m(x_1, \dots, x_m) = \mathrm{Res}_x(\mathrm{S}_{m-j}(x_1, \dots, x_{m-j-1}, x), \mathrm{S}_{j+2}(x_{m-j}, x_{m-j+1}, \dots, x_m, x))$$

where Res *denotes the resultant. For $m \geq 2$, the m-th summation polynomial is an irreducible symmetric polynomial that has degree 2^{m-2} in each of the variables.*

One can also give such polynomials in characteristic 2. For a general formula see Lemma 3.4 of [28]. For simplicity we consider the most important case of ordinary elliptic curves $E : y^2 + xy = x^3 + Ax^2 + B$. We have $\mathrm{S}_2(x_1, x_2) = x_1 - x_2$ as before and

$$\mathrm{S}_3(x_1, x_2, x_3) = (x_1x_2 + x_1x_3 + x_2x_3)^2 + x_1x_2x_3 + B.$$

The other formulae are generated using resultants in the same way.

Summation polynomials were proposed for index calculus algorithms by Semaev [107]. Remark 2 on the last page of [107] sketched an approach based on Weil descent that was fully developed by Gaudry [50] and Diem [28]. We sketch the current approach to these ideas.

We consider a discrete logarithm problem in an elliptic curve E defined over $\mathbb{F}_{q^n}$, where q is not necessarily a prime. We choose an $\mathbb{F}_q$-vector subspace V of $\mathbb{F}_{q^n}$ of dimension $1 \leq \ell < n$ and define the factor base to be

$$\mathcal{F} = \{P \in E(\mathbb{F}_{q^n}) : x(P) \in V\}. \tag{1}$$

Then one generates random points R as in Section 6 and tries to decompose over the factor base as $R = P_1 + \cdots + P_k$ with $P_i \in \mathcal{F}$. To solve the decomposition problem one computes the $(k+1)$-th summation polynomial and applies Weil restriction (Lemma 1) to the polynomial

$$\mathrm{S}_{k+1}(x_1, \dots, x_k, x(R)) = 0$$

to get a system of polynomials in $\mathbb{F}_q[y_{i,j}]$. One then adds linear constraints on the $y_{i,j}$ that corresponds to restricting the variables x_i to the vector space V. We have n equations in ℓk variables, and can try to solve using Gröbner basis methods. Note that solving systems of polynomial equations is a task of great difficulty, and estimating the complexity of such algorithms is non-trivial. Choosing ℓ and k is an important issue; in general we stick to $\ell k \approx n$ in order to have as many equations as indeterminates (in the next section, we discuss other choices).

A fundamental question is to determine the probability that a random point R can be written as a sum of k points chosen from the set $\mathcal{F}$ defined in equation (1). The first questions are whether such a set $\mathcal{F}$ is non-empty, and whether it can be expected to generate the group: see Propositions 4.11 and 4.29 of Diem [28], Kohel and Shparlinski [85], Kosters [86] and Shparlinski and Voloch [110] for some results on these questions. In practice we make the heuristic assumption that $\#\mathcal{F} \approx \#V$.

Dimension-1 factor base. When q is large and n is rather small, the best choice is $k = n$ and $\ell = 1$. In that case the factor base is built from a 1-dimensional $\mathbb{F}_q$-vector space V in $\mathbb{F}_{q^n}$ (as we will see, there are advantages from taking $V = \mathbb{F}_q$ to be a subfield). One can therefore view $\mathcal{F}$ as a variety of dimension 1 over $\mathbb{F}_q$, whose cardinality is around q. Taking symmetries into account, and assuming there are no unexpected cancellations, the number of elements that can be formed by summing up n elements of the factor basis should be roughly $\#\mathcal{F}^n/n! \approx q^n/n!$. Since the group size is roughly q^n we conclude that the proportion of decomposable elements is roughly $1/n!$. It remains to evaluate the cost of solving the polynomial system. Doing a heuristic complexity analysis for fixed n as in [50], we can assume that this takes polynomial time, so that all necessary relations can be computed in time $\tilde{O}(q)$, leading to a full algorithm with heuristic running time of $\tilde{O}(q^2)$. Using double-large-prime variation leads to an algorithm with heuristic time $\tilde{O}(q^{2-2/n})$. Several practical applications of this approach have been obtained and will be detailed in subsequent sections.

In [28], Diem did a careful study of how the cost of the polynomial system solving step grows (exponentially) with n; he was able to prove that there exists a sequence of prime powers $Q_i = q_i^{n_i}$ with $n_i \approx \sqrt{\log(q_i)}$ such that the ECDLP in $E(\mathbb{F}_{Q_i})$ can be solved (for any elliptic curve over $\mathbb{F}_{Q_i}$) in rigorous subexponential time $L_{Q_i}(2/3)$.

Higher dimension factor base. When n is not tiny, the $1/n!$ in the probability of success for the point decomposition becomes critical. This is the case for elliptic curves over $\mathbb{F}_{2^n}$ where n is prime, since the only proper subfield of $\mathbb{F}_{2^n}$ is $\mathbb{F}_q = \mathbb{F}_2$. Here, the only option is to take an $\mathbb{F}_q$-vector space $V \subseteq \mathbb{F}_{q^n}$ of dimension $\ell > 1$ to define the factor base. So $\mathcal{F}$ has cardinality roughly q^ℓ. The number of group elements formed as a sum of k factor base points is approximately $q^{k\ell}/k!$. Hence the probability of decomposition success is $\approx q^{\ell k - n}/k!$, which can be made high (but not too high, otherwise $\#\mathcal{F}$ is high as well, and then we need to collect too many relations). The main difficulty lies in solving the polynomial systems. Indeed, the number of variables is $k\ell \approx n$ which is not small, and the degree of the equations grows exponentially with k. In [29], Diem was able to improve the range of applicability of his rigorous subexponentiality result with this approach, allowing n up to $\log q$. These results (and others) shed no light on the most important case $E(\mathbb{F}_{2^n})$ with prime n. This latter case is the current subject of very active research. Section 10.2 is dedicated to it.

9 Point decomposition: Improvements and variants

We now discuss the problem of writing a random point R as a sum of k points from the factor base $\mathcal{F}$ of equation (1). As already mentioned, the standard way to do this is to apply Weil descent to the summation polynomial equation $\mathrm{S}_{k+1}(x_1, \ldots, x_k, x(R)) = 0$ to get a system of polynomial equations over a (relatively small) finite field $\mathbb{F}_q$.

According to [36, 77], the best approach to solve systems of polynomial equations is to compute a Gröbner basis with respect to the grevlex order and then apply the FGLM [34] algorithm or its sub-cubic variant [33] to transform the basis into a Gröbner basis with respect to the lexicographical order. The running time of the FGLM algorithm depends strongly on the number of solutions over the algebraic closure. The behaviour of the grevlex order computation is also affected by this quantity. Therefore, if q is not too large, it often makes sense to add the field equations $y_{i,j}^q - y_{i,j} = 0$ to the system, which guarantees that it defines a zero-dimensional algebraic set and reduces the number of solutions.

9.1 Symmetries

A simple observation is that the symmetric group $\mathcal{S}_m$ acts on a solution $(P_1, \ldots, P_m)$ to a point decomposition $R = P_1 + \cdots + P_m$. Consider the action of $\mathcal{S}_m$ on $\mathbb{F}_q[x_1, \ldots, x_m]$ given by $\sigma(x_i) = x_{\sigma(i)}$. If A is a ring and G is a finite group acting on A then we denote by $A^G = \{a \in A : \sigma(a) = a, \ \forall \sigma \in G\}$ the ring of invariants. It follows that $\mathrm{S}_{m+1}(x_1, \ldots, x_m, x_R) \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]^{\mathcal{S}_m}$. One can write down generators for the invariant ring (namely, the elementary symmetric polynomials in the x_i). With respect to these new variables the summation polynomial system has fewer solutions (the number of solutions is reduced by $m!$ in general) and potentially the polynomials have lower degree.

For example, take $m = 2$ and let $e_1 = x_1 + x_2$ and $e_2 = x_1 x_2$ be the first two elementary symmetric polynomials in $\{x_1, x_2\}$ so that $\mathbb{F}_{q^n}[x_1, x_2]^{\mathcal{S}_2} = \mathbb{F}_{q^n}[e_1, e_2]$. Using the formula for S_3 from Lemma 2 we have

$$\mathrm{S}_3(e_1, e_2, X_R) = (e_1^2 - 4e_2)X_R^2 - 2(e_1(e_2 + A_4) + 2A_6)X_R + ((e_2 - A_4)^2 - 4A_6 e_2).$$

For fixed X_R the total degree of the polynomial is lowered from 4 to 3, and the number of solutions is halved.

However, there is a serious issue that arises when one performs Weil descent and restricts the variables x_i to a vector subspace V. In the case where V is a subfield of $\mathbb{F}_{q^n}$ then the symmetric variables e_i are also constrained to V. Hence, it is natural to define the factor base in terms of $e_i \in V$ and the Weil descent process in terms of the new variables e_i proceeds in a straightforward way. More generally, as long as V is such that the set $V^{(2)} = \{v_1 v_2 \in \mathbb{F}_{q^n} : v_1, v_2 \in V\}$ is also a vector space over $\mathbb{F}_q$ of the same dimension as V then everything goes well.[2] However, if V is not closed under multiplication then the variables e_i when $i \geq 2$ may take values in a much larger set, and it is not so clear how to control the growth in variables as one performs the Weil restriction. This issue was discussed by Huang, Petit, Shinohara and Takagi [67] and they proposed to choose vector spaces V such that the $\mathbb{F}_q$-span of $V^{(i)} = \{v_1 \cdots v_i \in \mathbb{F}_{q^n} : v_1, \ldots, v_i \in V\}$ has as small dimension as possible. However, there is no doubt that the number of variables increases significantly when one does this.

To maximise the benefit from these ideas one wishes to find large groups that act on the summation polynomials. Faugère, Gaudry, Huot and Renault [36] considered the action of $[-1]$ (i.e., $P \mapsto -P$) and also the action $P \mapsto P + T$ where T is a point of order 2 or 4 (for twisted Edwards models in characteristic > 2). Hence, they study invariants under the action of a group of order $2^{m-1}m!$ or $4^{m-1}m!$. The paper [35] experiments further with these ideas and computes an 8-th summation polynomial in terms of invariant variables (previously the 8-th summation polynomial would have been unreachable). Vitse [113] did a more systematic study of which subgroups could be used in such a setting. Galbraith and Gebregiyorgis [43] considered the case of binary Edwards models over $E(\mathbb{F}_{2^n})$ where n is prime, combining the ideas of [67] with

[2] It is not necessary that V be a subfield. If V is a one-dimensional subspace that is not a subfield then $V^{(2)}$ is also a one-dimensional subspace, but $V^{(2)} \neq V$.

the use of invariants with respect to points of order 4. Even with all these ideas, the paper [43] concludes that these algorithmic ideas are not competitive with Pollard rho.

9.2 Unrolling the resultant

Recently a number of researchers [64, 68, 78, 106][3] have independently had an idea to lower the degree of the system of polynomial equations at the expense of more variables. This idea is sometimes called "unrolling the resultant" or "the splitting trick". It can be viewed as a special case of the idea of linearisation: replacing high degree monomials or polynomials with a new variable.

Recall from Lemma 2 that the summation polynomials are computed using the resultant as

$$\mathrm{S}_k(x_1, \ldots, x_k) = \mathrm{Res}_T(\mathrm{S}_{k-j}(x_1, \ldots, x_{k-j-1}, T), \mathrm{S}_{j+2}(x_{k-j}, x_{k-j+1}, \ldots, x_k, T)).$$

The resultant computation leads to exponential growth in the degree of the summation polynomials. Instead, one can use $k-2$ intermediate variables $T_i \in \mathbb{F}_{q^n}$ and consider the system

$$\begin{aligned} \mathrm{S}_3(x_1, x_2, T_1) &= 0 \\ \mathrm{S}_3(T_1, x_3, T_2) &= 0 \\ \vdots \quad &\vdots \\ \mathrm{S}_3(T_{k-2}, x_k, X_R) &= 0. \end{aligned}$$

Now take Weil descent from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q$ and impose the conditions $x_i \in V$ where V is an ℓ-dimensional $\mathbb{F}_q$-vector space. Recall that the original approach leads to a system of n polynomial equations in $k\ell$ variables. The new approach gives a system having $n(k-1)$ equations and $k\ell+(k-2)n$ variables (if $k\ell = n$ then both systems have the same number of equations as variables). The degree of the polynomial equations is dramatically lowered, but the number of variables is dramatically increased. Experimental results presented in [78, 106] suggest this idea can lead to improved running times, and [106] conjectures an algorithm with subexponential complexity. However, this conjecture is not yet demonstrated to our satisfaction (see Section 10.2 below).

Note that one can re-write the polynomials $\mathrm{S}_3(T_{j-1}, x_{j+1}, T_j)$ using symmetric variables with respect to $\mathcal{S}_3$, but it is an open problem to exploit larger symmetry groups in this situation.

9.3 Over- or under-determined systems

The point decomposition problem is, given R to find $P_1, \ldots, P_k$ in the factor base such that $R = P_1 + \cdots P_k$. We have seen that this can be converted to a system of polynomial equations, where one usually adjusts k to the smallest value such that the system admits solutions over the algebraic closure. In other words, the "natural" choice for k is the one that leads to a system with the same number of equations as

[3] And more, including the first author and his PhD student Shishay Gebregiyorgis.

unknowns. All the comparisons in this paragraph are with respect to this canonical situation, which is not always the best choice. Taking a smaller value of k, that is, having more equations than unknowns, is what is called an over-determined system, while the other choice is under-determined.

The main advantage of using over-determined systems is that the systems of polynomial equations have fewer variables and the degrees of the equations are lower. This is because the summation polynomial is of lower degree. The direct consequence is that solving these systems is much faster; in fact, there are cases where it drops from completely infeasible to feasible with off-the-shelf tools. Of course, this does not come for free. The success probability of the decomposition is divided by the cardinality of the factor base each time the value of k is decremented. There is a trade-off, and it does not always pay to use over-determined systems. In [77], Joux and Vitse give a theoretical study of this trade-off, and a practical application to an elliptic curve over an extension of degree 5. A particular context where this approach can be very efficient is an oracle-assisted attack on the static Diffie-Hellman problem; indeed, in that case, only one successful decomposition is needed, and we can afford a huge loss in the probability of success [55, 56].

The opposite approach is to increase k and get an under-determined system. Since there are more unknowns than constraints, it is a system of positive dimension; one can also view it as a 0-dimensional system over a rational function field. For instance, let us consider the x-coordinate x_k of P_k as a parameter. Then the result of the Gröbner basis is a generic system parametrized by x_k, such that pluging any value for x_k in it, we obtain directly[4] the Gröbner basis corresponding to the decomposition of $R - P_k$. The advantage is immediately visible: for the price of one Gröbner basis computation, we get many for free. But there is again a trade-off: the generic Gröbner basis is much more costly than a single standard one. It could well be that it is impossible to compute it. To complicate the analysis, working with a generic Gröbner basis opens the way to sieving, as demonstrated by Joux and Vitse [76], which can have a huge practical impact. In fact, in most cases, the generic Gröbner basis is too large, and this technique does not pay. The most notable exception currently found in the literature is using Nagao's method (for hyperelliptic curves of genus > 1) that we present now.

9.4 Nagao's approach

In the context of computing discrete logarithms in hyperelliptic curves over extension fields, there is no known equivalent to Semaev's summation polynomials. This was the motivation for Nagao [95] to introduce another approach to solve the point decomposition problem faster than by using the system of equations coming from a naive application of the group law.

The idea is to read the decomposition equation $R - (P_1 + \cdots + P_k) = 0$ in terms of principal divisors. Riemann-Roch theory dictates the form of the corresponding function, even if the P_i's are not yet known: it must be a linear combination of a few easily-computable functions. The solving strategy is now clear: put indeterminates for the coefficients of the linear combination and consider the conditions for the P_i to be in the factor base. As with summation polynomials, this translates into a system of polynomial equations that can be solved with Gröbner basis computations. A nice

[4] This is true only under genericity assumptions, and with appropriate monomial orderings.

feature of this system is that all the equations are quadratic. The approach is quite general: it can be extended to more general curves, although the equations are no longer quadratic [96].

Nagao's approach is also valid for elliptic curves. However, until now, nobody managed to turn this into an algorithm that is faster than an approach based on summation polynomials. A recent remark in [104] shows however that this could be used to produce relations almost for free in the case of elliptic curves over extension fields, although not in a large enough quantity: other relations need to be found in another, more costly, way.

9.5 Mixing everything

For a given discrete logarithm problem, several of the above techniques can be mixed. For instance, Nagao's approach can be used with under-determined systems, and then sieving can be used.

The most impressive combination of techniques has been done by Joux and Vitse [76] for attacking a 149-bit subgroup of an elliptic curve over $\mathbb{F}_{p^6}$. They first used a GHS technique to reduce to a discrete logarithm problem in a genus-3 hyperelliptic curve over $\mathbb{F}_{p^2}$, and then they used an under-determined Nagao-like point decomposition approach and sieving. We remark that their fast technique naturally constructs relations that do not involve the input point, so they used a "right-kernel linear algebra" approach as described in Section 6.2.

10 Further open questions

10.1 Ideas from the finite field DLP

Joux and others have recently made tremendous progress on the DLP in $\mathbb{F}_{q^n}^*$ where q is a small prime power and $n \approx q$ is large (see the survey paper in this volume). A key idea is to exploit the equation

$$x^q - x \;=\; \prod_{\alpha \in \mathbb{F}_q} (x - \alpha) \tag{2}$$

in $\mathbb{F}_q[x]$. This equation is interpreted in two ways: the left hand side is a polynomial to be factored and the right hand side is a product of factor base elements. One also chooses a field representation so that the Frobenius x^q is represented as a rational function of low degree. Another key idea is to substitute into equation (2) a Möbius transform $(ax+b)/(cx+d)$ with $a, b, c, d \in \mathbb{F}_{q^d}$ for some small $d > 1$. This is based on the fact that the automorphisms of the projective line are given by Möbius transformations.

It is tempting to seek an elliptic curve version of these ideas. One natural approach is to use the double cover $E \to \mathbb{P}^1$ given by $(x, y) \mapsto x$. Pulling back equation (2) gives

$$\operatorname{div}(x^q - x) \;=\; \sum_{P \in E(\overline{\mathbb{F}}_q), x(P) \in \mathbb{F}_q} (P) \;-\; 2q(\infty).$$

However, this is un-interesting from a group-theoretic viewpoint, as the right hand side just represents the fact that $x(P) = x(-P)$ and $P + (-P) = \infty$. Some further obstacles to the method are that $\mathrm{Aut}(E)$ is very small compared with $\mathrm{Aut}(\mathbb{P}^1)$. We also have no idea how to do the individual discrete log stage.

More generally, one can wonder whether some quasi-polynomial discrete logarithm algorithm can be found for curves of non-zero genus. A study by Massierer [90] seems to imply that the only curves for which one can find relations quickly are very special, like the famous Hermitian curve. This is still quite a mystery, though.

10.2 A subexponential algorithm for elliptic curves over $\mathbb{F}_{2^n}$?

The big question at the time of writing this article is whether or not the strategies described in the previous sections can lead to a general subexponential discrete logarithm algorithm for elliptic curves in characteristic 2. The key point that would lead to such a major result is that the polynomial systems arising from the decomposition problem after Weil restriction are not generic. It was shown in [37] that their special structure makes them easier to solve (but in that paper the overall algorithm is not faster than a generic discrete logarithm algorithm).

An approach that looks promising is to study the so-called first-fall degree (FFD) of the polynomial systems. A Gröbner-basis algorithm like F4 or F5 proceeds with polynomials of increasing degree while computing the basis, maybe backtracking to handle smaller degree polynomials if they happen to be created along the process. Roughly speaking, the FFD is the degree currently handled by the algorithm when this backtrack occurs for the first time. It has been observed that this degree is often not too far from the maximal degree that will ever occur in the algorithm. This is the first fall degee assumption (FFDA). Therefore, being able to bound the FFD, and assuming that the FFDA holds for the systems involved in ECDLP, we could dream to get better estimates for the whole computation.

There has been a line of research in this direction [100, 108, 67, 62, 68, 78, 106, 87], where bounds have been proven for the FFD and/or experiments have been performed to give evidence to support the FFDA. None of this research definitely settled the question of subexponentiality, but the best results so far have been obtained with the "unrolling resultant" technique (see Section 9.2). Depending on the variant for solving the decomposition problem and how much one is ready to believe, this has lead some researchers to conjecture a subexponential complexity for the ECDLP in $E(\mathbb{F}_{2^n})$ of $L_{2^n}(2/3)$ or even $L_{2^n}(1/2)$.

After some experiments that tended to raise some doubts about the subexponentiality claims, another notion has been recently introduced in an attempt to help the analysis: the last fall degree (LFD) of a polynomial system [66, 64]. The main advantage is that this is a well defined notion that does not depend on a particular system solving algorithm, nor on the monomial ordering chosen for this solving. Furthermore, there exists a solving algorithm with a complexity that can be bounded using the LFD as a parameter. In terms of complexity estimates, it seems as good as any known Gröbner basis or XL-like algorithm. The drawback is that the LFD is much harder to bound than the FFD for systems related to ECDLP.

The current situation, not at all definitive, is that there is no consensus whether there is a subexponential algorithm for ECDLP in characteristic 2. The FFDA approach seems to be too optimistic, while the LFD approach looks more precise, but much harder to estimate.

The future lines of research on this topic might be:

- Prove the LFD for polynomial systems coming from the ECDLP;
- Find a new notion to help the analysis;
- Find a new strategy for solving the decomposition problem that helps the analysis.

Finally, it must be emphasized that for the moment none of the approaches are practical at all: even with the most optimistic assumptions, the running time and the memory usage would be extremely high for any key size currently in use. The difficulty to make practical experiments with non-tiny examples is an explanation why the asumptions are hard to (in-)validate[5].

10.3 Breaking symmetries

When computing relations of the form $R = P_1 + \cdots + P_k$ there is an inevitable factor of $k!$ in the success probability. One suggestion to mitigate this problem (according to Section 7 of Nagao [96], Matsuo was the first to consider this idea) is to use k disjoint factor bases and require $P_i \in \mathcal{F}_i$. One avoids the $k!$ factor, but the linear algebra problem is increased by a factor of k, so one needs k times as many relations. Nevertheless, one might therefore expect an overall speedup by a factor of $(k-1)!$. A detailed analysis is given by Galbraith and Gebregiyorgis [43], where it is explained how to combine the approach with invariant coordinates for summation polynomials.

Diem [29] has also used disjoint factor bases, but for other reasons.

10.4 Subfield curves

A major open problem is to speed-up index calculus algorithms for the case of subfield curves. Ideally one wishes to use the Frobenius endomorphism to either increase the probability of successfully finding a relation or else to have a method to generate lots of relations from each solution of a system of polynomial equations.

Gorla and Massierer [54] have given an interesting approach to this question. They represent the ECDLP instance on a subfield curve using the trace zero variety. Then they perform an index calculus method using summation polynomials directly on the trace zero variety. This approach does not currently lead to a dramatic speed-up.

Acknowledgements We thank Claus Diem, Michiel Kosters, Christophe Petit, Peter Wild and an anonymous referee for helpful comments on the draft of this article. The second author also thanks Maike Massierer, Pierre-Jean Spaenlehauer and Vanessa Vitse for various discussions on the topic.

References

1. Adleman, L., DeMarrais, J., Huang, M.D.: A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In: L.M. Adleman, M.D. Huang (eds.) ANTS I, *LNCS*, vol. 877, pp. 28–40. Springer (1994)
2. Augot, D., Morain, F.: Discrete logarithm computations over finite fields using Reed-Solomon codes. arXiv:1202.4361 (2012)

[5] And one must be careful not to be fooled by the *Strong law of small numbers* [57].

3. Avanzi, R., Cohen, H., Doche, C., Frey, G., Lange, T., Nguyen, K., Vercauteren, F.: Handbook of elliptic and hyperelliptic cryptography. Chapman and Hall/CRC (2006)
4. Babai, L., Szemerédi, E.: On the complexity of matrix group problems I. Foundations of Computer Science (FOCS) pp. 229–240 (1996)
5. Bailey, D.V., Batina, L., Bernstein, D.J., Birkner, P., Bos, J.W., Chen, H.C., Cheng, C.M., van Damme, G., de Meulenaer, G., Perez, L.J.D., Fan, J., Güneysu, T., Gurkaynak, F., Kleinjung, T., Lange, T., Mentens, N., Niederhagen, R., Paar, C., Regazzoni, F., Schwabe, P., Uhsadel, L., Herrewege, A.V., Yang, B.Y.: Breaking ECC2K-130. Cryptology ePrint Archive: Report 2009/541 (2009). URL `http://ecc-challenge.info/`
6. Bernstein, D.J., Lange, T.: Computing small discrete logarithms faster. In: S.D. Galbraith, M. Nandi (eds.) INDOCRYPT 2012, *LNCS*, vol. 7668, pp. 317–338. Springer (2012)
7. Bernstein, D.J., Lange, T.: Non-uniform cracks in the concrete: The power of free precomputation. In: K. Sako, P. Sarkar (eds.) ASIACRYPT 2013, *LNCS*, vol. 8270, pp. 321–340. Springer (2013)
8. Bernstein, D.J., Lange, T.: Two grumpy giants and a baby. In: E.W. Howe, K.S. Kedlaya (eds.) Proceedings of the Tenth Algorithmic Number Theory Symposium, *Open Book Series*, vol. 1, pp. 87–111. MSP (2013)
9. Bernstein, D.J., Lange, T., Farashahi, R.R.: Binary Edwards curves. In: E. Oswald, P. Rohatgi (eds.) CHES 2008, *LNCS*, vol. 5154, pp. 244–265. Springer (2008)
10. Bernstein, D.J., Lange, T., Schwabe, P.: On the correct use of the negation map in the Pollard rho method. In: D. Catalano, N. Fazio, R. Gennaro, A. Nicolosi (eds.) PKC 2011, *LNCS*, vol. 6571, pp. 128–146. Springer (2011)
11. Blackburn, S.R., Murphy, S.: The number of partitions in Pollard rho. Unpublished manuscript (1998)
12. Blake, I.F., Seroussi, G., Smart, N.P.: Elliptic Curves in Cryptography. Cambridge (1999)
13. Blake, I.F., Seroussi, G., Smart, N.P.: Advances in Elliptic Curve Cryptography. Cambridge (2005)
14. Boneh, D., Boyen, X.: Short signatures without random oracles. In: C. Cachin, J. Camenisch (eds.) EUROCRYPT 2004, *LNCS*, vol. 3027, pp. 56–73. Springer (2004)
15. Bos, J.W., Costello, C., Miele, A.: Elliptic and hyperelliptic curves: A practical security analysis. In: H. Krawczyk (ed.) PKC 2014, *LNCS*, vol. 8383, pp. 203–220. Springer (2014)
16. Bos, J.W., Kaihara, M.E., Kleinjung, T., Lenstra, A.K., Montgomery, P.L.: Solving a 112-bit prime elliptic curve discrete logarithm problem on game consoles using sloppy reduction. IJACT **2**(3), 212–228 (2012)
17. Bos, J.W., Kleinjung, T., Lenstra, A.K.: On the use of the negation map in the Pollard Rho method. In: G. Hanrot, F. Morain, E. Thomé (eds.) ANTS IX, *LNCS*, vol. 6197, pp. 66–82. Springer (2010)
18. Brown, D.R.L., Gallant, R.P.: The static Diffie-Hellman problem. Cryptology ePrint Archive: Report 2004/306 (2004)
19. Certicom Research: Certicom ECC challenge. URL `https://www.certicom.com/images/pdfs/challenge-2009.pdf`. Updated in November 10, 2009
20. Chateauneuf, M., Ling, A.C.H., Stinson, D.R.: Slope packings and coverings, and generic algorithms for the discrete logarithm problem. J. Combinatorial Designs

11(1), 36–50 (2003)

21. Cheng, Q.: Hard problems of algebraic geometry codes. IEEE Trans. Information Theory **54**(1), 404–406 (2008)
22. Cheon, J.H.: Security analysis of the strong Diffie-Hellman problem. In: S. Vaudenay (ed.) EUROCRYPT 2006, *LNCS*, vol. 4004, pp. 1–11. Springer (2006)
23. Cheon, J.H.: Discrete logarithm problem with auxiliary inputs. J. Cryptology **23**(3), 457–476 (2010)
24. Cheon, J.H., Kim, T., Song, Y.S.: A group action on $\mathbb{Z}_p^*$ and the generalized DLP with auxiliary inputs. In: T. Lange, K.E. Lauter, P. Lisonek (eds.) SAC 2013, *LNCS*, vol. 8282, pp. 121–135. Springer (2014)
25. Diem, C.: The GHS-attack in odd characteristic. J. Ramanujan Math. Soc. **18**(1), 1–32 (2003)
26. Diem, C.: An index calculus algorithm for plane curves of small degree. In: F. Hess, S. Pauli, M.E. Pohst (eds.) ANTS VII, *LNCS*, vol. 4076, pp. 543–557. Springer (2006)
27. Diem, C.: On the discrete logarithm problem in class groups of curves. Math. Comp. **80**(273), 443–475 (2011)
28. Diem, C.: On the discrete logarithm problem in elliptic curves. Compositio Math. **147**, 75–104 (2011)
29. Diem, C.: On the discrete logarithm problem in elliptic curves II. Algebra and Number Theory **7**(6), 1281–1323 (2013)
30. Diem, C., Kochinke, S.: Computing discrete logarithms with special linear systems (2013). Preprint
31. Diem, C., Scholten, J.: Cover attacks - a report for the AREHCC project (2003). Preprint
32. Driencourt, Y., Michon, J.F.: Elliptic codes over fields of characteristics 2. J. Pure Appl. Algebra **45**(1), 15–39 (1987)
33. Faugère, J., Gaudry, P., Huot, L., Renault, G.: Sub-cubic change of ordering for Gröbner basis: a probabilistic approach. In: ISSAC 2014, pp. 170–177. ACM (2014)
34. Faugère, J., Gianni, P., Lazard, D., Mora, T.: Efficient computation of zero-dimensional Gröbner bases by change of ordering. Journal of Symbolic Computation **16**(4), 329–344 (1993)
35. Faugère, J., Huot, L., Joux, A., Renault, G., Vitse, V.: Symmetrized summation polynomials: Using small order torsion points to speed up elliptic curve index calculus. In: P.Q. Nguyen, E. Oswald (eds.) EUROCRYPT 2014, *LNCS*, vol. 8441, pp. 40–57. Springer (2014)
36. Faugère, J.C., Gaudry, P., Huot, L., Renault, G.: Using symmetries in the index calculus for elliptic curves discrete logarithm. J. Cryptology **27**(4), 595–635 (2014)
37. Faugère, J.C., Perret, L., Petit, C., Renault, G.: Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In: D. Pointcheval, T. Johansson (eds.) EUROCRYPT 2012, *LNCS*, vol. 7237, pp. 27–44. Springer (2012)
38. Fouque, P., Joux, A., Mavromati, C.: Multi-user collisions: Applications to discrete logarithm, Even-Mansour and PRINCE. In: P. Sarkar, T. Iwata (eds.) ASIACRYPT 2014, *LNCS*, vol. 8873, pp. 420–438. Springer (2014)
39. Frey, G.: Applications of arithmetic geometry to cryptographic constructions. In: D. Jungnickel, N. Niederreiter (eds.) Finite Fields and Applications, pp. 128–161. Springer (2001)

40. Frey, G.: On the relation between Brauer groups and discrete logarithms. Tatra Mt. Math. Publ. **35**, 1–29 (2006)
41. Galbraith, S.D.: Constructing isogenies between elliptic curves over finite fields. LMS J. Comput. Math. **2**, 118–138 (1999)
42. Galbraith, S.D.: Mathematics of public key cryptography. Cambridge University Press (2012)
43. Galbraith, S.D., Gebregiyorgis, S.W.: Summation polynomial algorithms for elliptic curves in characteristic two. In: W. Meier, D. Mukhopadhyay (eds.) INDOCRYPT 2014, *LNCS*, vol. 8885, pp. 409–427. Springer (2014)
44. Galbraith, S.D., Hess, F., Smart, N.P.: Extending the GHS Weil descent attack. In: L.R. Knudsen (ed.) EUROCRYPT 2002, *LNCS*, vol. 2332, pp. 29–44. Springer (2002)
45. Galbraith, S.D., Pollard, J.M., Ruprai, R.S.: Computing discrete logarithms in an interval. Math. Comp. **82**(282), 1181–1195 (2013)
46. Galbraith, S.D., Ruprai, R.S.: Using equivalence classes to accelerate solving the discrete logarithm problem in a short interval. In: P.Q. Nguyen, D. Pointcheval (eds.) PKC 2010, *LNCS*, vol. 6056, pp. 368–383. Springer (2010)
47. Galbraith, S.D., Smart, N.P.: A cryptographic application of Weil descent. In: M. Walker (ed.) IMA Cryptography and Coding, *LNCS*, vol. 1746, pp. 191–200. Springer (1999)
48. Galbraith, S.D., Wang, P., Zhang, F.: Computing elliptic curve discrete logarithms with improved baby-step giant-step algorithm. Preprint (2015)
49. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Improving the parallelized Pollard lambda search on binary anomalous curves. Math. Comp. **69**(232), 1699–1705 (2000)
50. Gaudry, P.: Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. J. Symbolic Computation **44**(12), 1690–1702 (2009)
51. Gaudry, P., Hess, F., Smart, N.P.: Constructive and destructive facets of Weil descent on elliptic curves. J. Cryptology **15**(1), 19–46 (2002)
52. Gaudry, P., Schost, É.: A low-memory parallel version of Matsuo, Chao, and Tsujii's algorithm. In: D.A. Buell (ed.) ANTS VI, *LNCS*, vol. 3076, pp. 208–222. Springer (2004)
53. Gaudry, P., Thomé, E., Thériault, N., Diem, C.: A double large prime variation for small genus hyperelliptic index calculus. Math. Comp. **76**(257), 475–492 (2007)
54. Gorla, E., Massierer, M.: Index calculus in the trace zero variety. Cryptology ePrint Archive: Report 2014/318 (2014). To appear in Advances in Mathematics of Communications
55. Granger, R.: On the static Diffie-Hellman problem on elliptic curves over extension fields. In: M. Abe (ed.) ASIACRYPT 2010, *Lecture Notes in Computer Science*, vol. 6477, pp. 283–302. Springer (2010)
56. Granger, R., Joux, A., Vitse, V.: New timings for oracle-assisted SDHP on the IPSEC Oakley "well known group" 3 curve. Announcement on the NMBRTHRY mailing list, July 2010
57. Guy, R.K.: The strong law of small numbers. American Mathematical Monthly **95**(8), 697–712 (1988)
58. Hankerson, D., Menezes, A., Vanstone, S.: Guide to elliptic curve cryptography. Springer (2004)

59. Hess, F.: Computing relations in divisor class groups of algebraic curves over finite fields (2003). Preprint
60. Hess, F.: Generalising the GHS attack on the elliptic curve discrete logarithm problem. LMS Journal of Computation and Mathematics **7**, 167–192 (2004)
61. Hitchcock, Y., Montague, P., Carter, G., Dawson, E.: The efficiency of solving multiple discrete logarithm problems and the implications for the security of fixed elliptic curves. Int. J. Inf. Secur. **3**, 86–98 (2004)
62. Hodges, T.J., Petit, C., Schlather, J.: First fall degree and Weil descent. Finite Fields and Their Applications **30**, 155–177 (2014)
63. Hong, J., Lee, H.: Analysis of possible pre-computation aided DLP solving algorithms. J. Korean Math. Soc. **52**(4), 797–819 (2015)
64. Huang, M.A., Kosters, M., Yeo, S.L.: Last fall degree, HFE, and Weil descent attacks on ECDLP. In: R. Gennaro, M. Robshaw (eds.) CRYPTO 2015, *LNCS*, vol. 9215, pp. 581–600. Springer (2015)
65. Huang, M.D., Raskind, W.: Global duality, signature calculus and the discrete logarithm problem. LMS Journal of Computation and Mathematics **12**, 228–263 (2009)
66. Huang, M.D.A., Kosters, M., Yang, Y., Yeo, S.L.: On the last fall degree of zero-dimensional Weil descent systems (2015). ArXiv preprint 1505.02532
67. Huang, Y., Petit, C., Shinohara, N., Takagi, T.: Improvement of Faugère et al.'s method to solve ECDLP. In: K. Sakiyama, M. Terada (eds.) IWSEC 2013, *LNCS*, vol. 8231, pp. 115–132. Springer (2013)
68. Huang, Y., Petit, C., Shinohara, N., Takagi, T.: On generalized first fall degree assumptions. Cryptology ePrint Archive: Report 2015/358 (2015)
69. Hyung Tae Lee Jung Hee Cheon, J.H.: Accelerating ID-based encryption based on trapdoor DL using pre-computation. Cryptology ePrint Archive: Report 2011/187 (2011)
70. Iijima, T., Momose, F., Chao, J.: A classification of elliptic curves with respect to the GHS attack in odd characteristic (2015). Cryptology ePrint Archive: Report 2015/805
71. J.-H. Kim R. Montenegro, Y.P., Tetali, P.: A birthday paradox for Markov chains, with an optimal bound for collision in the Pollard rho algorithm for discrete logarithm. The Annals of Applied Probability **20**(2), 295–521 (2010)
72. Jacobson Jr., M.J., Koblitz, N., Silverman, J.H., Stein, A., Teske, E.: Analysis of the Xedni calculus attack. Des. Codes Crypt. **20**(1), 41–64 (2000)
73. Jao, D., Miller, S.D., Venkatesan, R.: Do all elliptic curves of the same order have the same difficulty of discrete log? In: B.K. Roy (ed.) ASIACRYPT 2005, *LNCS*, vol. 3788, pp. 21–40. Springer (2005)
74. Joux, A.: Algorithmic cryptanalysis. Chapman & Hall/CRC (2009)
75. Joux, A., Lercier, R., Naccache, D., Thomé, E.: Oracle-assisted static Diffie-Hellman is easier than discrete logarithms. In: M.G. Parker (ed.) Cryptography and Coding, 12th IMA International Conference, *LNCS*, vol. 5921, pp. 351–367. Springer (2009)
76. Joux, A., Vitse, V.: Cover and decomposition index calculus on elliptic curves made practical – Application to a previously unreachable curve over $\mathbb{F}_{p^6}$. In: Advances in Cryptology – EUROCRYPT 2012, *LNCS*, vol. 7237, pp. 9–26. Springer (2012)
77. Joux, A., Vitse, V.: Elliptic curve discrete logarithm problem over small degree extension fields – Application to the static Diffie-Hellman problem on $E(\mathbb{F}_{q^5})$. J.

Cryptology **26**(1), 119–143 (2013)

78. Karabina, K.: Point decomposition problem in binary elliptic curves. Cryptology ePrint Archive: Report 2015/319 (2015)
79. Kijima, S., Montenegro, R.: Collision of random walks and a refined analysis of attacks on the discrete logarithm problem. In: J. Katz (ed.) PKC 2015, *LNCS*, vol. 9020, pp. 127–149. Springer (2015)
80. Kim, J.H., Montenegro, R., Tetali, P.: Near optimal bounds for collision in Pollard rho for discrete log. In: Foundations of Computer Science (FOCS), pp. 215–223. IEEE (2007)
81. Kim, M., Cheon, J.H., Lee, I.S.: Analysis on a generalized algorithm for the strong discrete logarithm problem with auxiliary inputs. Math. Comp. **83**(288), 1993–2004 (2014)
82. Kim, T., Cheon, J.H.: A new approach to the discrete logarithm problem with auxiliary inputs. Cryptology ePrint Archive: Report 2012/609 (2012)
83. Koblitz, N., Menezes, A.: Another look at non-standard discrete log and Diffie-Hellman problems. J. Mathematical Cryptology **2**(4), 311–326 (2008)
84. Koblitz, N., Menezes, A.: Intractable problems in cryptography. In: G. McGuire, G.L. Mullen, D. Panario, I.E. Shparlinski (eds.) Finite Fields: Theory and Applications, *Contemporary Mathematics*, vol. 518, pp. 279–300. AMS (2010)
85. Kohel, D.R., Shparlinski, I.E.: On exponential sums and group generators for elliptic curves over finite fields. In: W. Bosma (ed.) ANTS IV, *LNCS*, vol. 1838, pp. 395–404. Springer (2000)
86. Kosters, M.: Deterministically generating Picard groups of hyperelliptic curves over finite fields (2014). ArXiv preprint 1402.6579
87. Kosters, M., Yeo, S.L.: Notes on summation polynomials (2015). ArXiv preprint 1503.08001
88. Kozaki, S., Kutsuma, T., Matsuo, K.: Remarks on Cheon's algorithms for pairing-related problems. In: T. Takagi, T. Okamoto, E. Okamoto, T. Okamoto (eds.) Pairing 2007, *LNCS*, vol. 4575, pp. 302–316. Springer (2007)
89. Kuhn, F., Struik, R.: Random walks revisited: Extensions of Pollard's rho algorithm for computing multiple discrete logarithms. In: S. Vaudenay, A.M. Youssef (eds.) SAC 2001, *LNCS*, vol. 2259, pp. 212–229. Springer (2001)
90. Massierer, M.: Some experiments investigating a possible $L(1/4)$ algorithm for the discrete logarithm problem in algebraic curves (2014). Cryptology ePrint Archive: Report 2014/996
91. May, A., Ozerov, I.: A generic algorithm for small weight discrete logarithms in composite groups. In: A. Joux, A.M. Youssef (eds.) SAC 2014, *LNCS*, vol. 8781, pp. 278–289. Springer (2014)
92. Menezes, A., Qu, M.: Analysis of the Weil descent attack of Gaudry, Hess and Smart. In: D. Naccache (ed.) CT-RSA 2001, *LNCS*, vol. 2020, pp. 308–318. Springer (2001)
93. Momose, F., Chao, J.: Elliptic curves with weak coverings over cubic extensions of finite fields with odd characteristics. J. Ramanujan Mathematical Society **28**(3), 299–357 (2013)
94. Montenegro, R., Tetali, P.: How long does it take to catch a wild kangaroo? In: Symposium on Theory of Computing (STOC), pp. 553–559 (2009)
95. Nagao, K.i.: Decomposition attack for the Jacobian of a hyperelliptic curve over an extension field. In: G. Hanrot, F. Morain, E. Thomé (eds.) ANTS-IX: Algorithmic Number Theory, *LNCS*, vol. 6197, pp. 285–300. Springer (2010)

96. Nagao, K.i.: Decomposition formula of the Jacobian group of plane curve (2013). Cryptology ePrint Archive: Report 2013/548
97. Nechaev, V.I.: Complexity of a determinate algorithm for the discrete logarithm. Mathematical Notes **55**(2), 165–172 (1994)
98. Nguyen, K.: Explicit arithmetic of Brauer groups, ray class fields and index calculus. PhD thesis, University Essen (2001)
99. Oorschot, P., Wiener, M.J.: Parallel collision search with cryptanalytic applications. J. Cryptology **12**(1), 1–28 (1999)
100. Petit, C., Quisquater, J.J.: On polynomial systems arising from a Weil descent. In: X. Wang, K. Sako (eds.) ASIACRYPT 2012, *LNCS*, vol. 7658, pp. 451–466. Springer (2012)
101. Pollard, J.M.: Kangaroos, Monopoly and discrete logarithms. J. Cryptology **13**(4), 437–447 (2000)
102. Pomerance, C.: Fast, rigorous factorization and discrete logarithm algorithms. In: D.S. Johnson, T. Nishizeki, A. Nozaki, H.S. Wolf (eds.) Discrete Algorithms and Complexity, Proceedings of the Japan–US Joint Seminar, June 4–6, 1986, Kyoto, Japan, Perspectives in Computing, pp. 119–143. Academic Press, Orlando (1987)
103. Sakemi, Y., Hanaoka, G., Izu, T., Takenaka, M., Yasuda, M.: Solving a discrete logarithm problem with auxiliary input on a 160-bit elliptic curve. In: M. Fischlin, J.A. Buchmann, M. Manulis (eds.) PKC 2012, *LNCS*, vol. 7293, pp. 595–608. Springer (2012)
104. Sarkar, P., Singh, S.: A simple method for obtaining relations among factor basis elements for special hyperelliptic curves (2015). Cryptology ePrint Archive: Report 2015/179
105. Satoh, T.: On generalization of Cheon's algorithm. Cryptology ePrint Archive: Report 2009/058 (2009)
106. Semaev, I.: New algorithm for the discrete logarithm problem on elliptic curves. Cryptology ePrint Archive: Report 2015/310 (2015)
107. Semaev, I.A.: Summation polynomials and the discrete logarithm problem on elliptic curves. Cryptology ePrint Archive: Report 2004/031 (2004)
108. Shantz, M., Teske, E.: Solving the elliptic curve discrete logarithm problem using Semaev polynomials, Weil descent and Gröbner basis methods – an experimental study. In: Number Theory and Cryptography, *LNCS*, vol. 8260, pp. 94–107. Springer (2013)
109. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: W. Fumy (ed.) EUROCRYPT 1997, *LNCS*, vol. 1233, pp. 256–266. Springer (1997)
110. Shparlinski, I.E., Voloch, J.F.: Generators of elliptic curves over finite fields. Bull. Inst. Math. Acad. Sinica. **9**(4), 657–670 (2014)
111. Thériault, N.: Index calculus attack for hyperelliptic curves of small genus. In: C.S. Laih (ed.) ASIACRYPT 2003, *LNCS*, vol. 2894, pp. 75–92. Springer (2003)
112. Thériault, N.: Weil descent attack for Kummer extentions. J. Ramanujan Mathematical Society **18**(3), 281–312 (2003)
113. Vitse, V.: Summation polynomials and symmetries for the ECDLP over extension fields. Talk given at the DLP 2014 workshop, Ascona (2014)
114. Washington, L.C.: Elliptic Curves: Number Theory and Cryptography, 2nd edn. CRC Press (2008)
115. Wenger, E., Wolfger, P.: Solving the discrete logarithm of a 113-bit Koblitz curve with an FPGA cluster. In: A. Joux, A.M. Youssef (eds.) SAC 2014, *LNCS*, vol.

8781, pp. 363–379. Springer (2014)

116. Wenger, E., Wolfger, P.: Harder, better, faster, stronger – elliptic curve discrete logarithm computations on FPGAs. Cryptology ePrint Archive: Report 2015/143 (2015)
117. Wiener, M.J., Zuccherato, R.J.: Faster attacks on elliptic curve cryptosystems. In: S.E. Tavares, H. Meijer (eds.) SAC 1998, *LNCS*, vol. 1556, pp. 190–200. Springer (1998)
118. Yun, A.: Generic hardness of the multiple discrete logarithm problem. In: E. Oswald, M. Fischlin (eds.) EUROCRYPT 2015, *LNCS*, vol. 9057, pp. 817–836. Springer (2015)
119. Zhang, F., Wang, P.: Speeding up elliptic curve discrete logarithm computations with point halving. Des. Codes Crypt. **67**(2), 197–208 (2013)