On the discrete logarithm problem for prime-field elliptic curves

Alessandro Amadori[a], Federico Pintore[b,1,*], Massimiliano Sala[b,2]

[a] *Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*
[b] *Department of Mathematics, University of Trento, 38123 Trento, Italy*

Abstract

In recent years several papers have appeared that investigate the classical discrete logarithm problem for elliptic curves by means of the multivariate polynomial approach based on the celebrated summation polynomials, introduced by Semaev in 2004. With a notable exception by Petit *et al.* in 2016, all numerous papers on the subject have investigated only the composite-field case, leaving apart the laborious prime-field case. In this paper we propose a variation of Semaev's original approach that reduces to only one the relations to be found among points of the factor base, thus decreasing drastically the necessary Groebner basis computations. Our proposal holds for any finite field but it is particularly suitable for the prime-field case, where it outperforms both the original Semaev's method and the specialised algorithm by Petit *et al.*.

Keywords: Elliptic curve, Discrete logarithm problem (DLP), Prime field, Summation polynomials, Groebner basis
2010 MSC: 11T71, 11Y16, 11G20, 13P10, 14G15, 14H52

1. Introduction

Several cryptographic schemes base their security upon the hardness of the discrete logarithm problem for elliptic curves (ECDLP) [13, 15]. For an elliptic curve E defined over a finite field $\mathbb{K}$, an instance of the ECDLP is the following:

given $P, Q \in E(\mathbb{K})$, compute an integer w, if it exists, s.t. $Q = wP$.

The best known algorithms for the ECDLP are algorithms that work on arbitrary cyclic groups - like Pollard's Rho algorithm [19], which runs in time $\mathcal{O}(\sqrt{|E(\mathbb{K})|})$ if $|E(\mathbb{K})|$ is prime - exception made for algorithms that are specific for some families of weak curves (see, for example, [14]).

In 2004 Semaev introduced [20] a family of polynomials, named *summation polynomials*, proposing their exploitation for an index calculus algorithm for elliptic curves. The Index Calculus is originally a subexponential algorithm to compute discrete logarithms in the multiplicative groups of finite fields. However, it is customary to use the name *index calculus algorithm* to refer to any algorithm that computes discrete logarithms in a cyclic group G by first collecting linear relations and, afterwards, using linear algebra. Following [6] and restricting to the case $G = E(\mathbb{K})$, with $r = |E(\mathbb{K})|$ a prime integer, the simplest version of index

*Corresponding author
Email addresses: a.amadori@tue.nl (Alessandro Amadori), federico.pintore@gmail.com (Federico Pintore), maxsalacodes@gmail.com (Massimiliano Sala)

[1] This work was supported by CARITRO Foundation [grant number: 2014.0500]
[2] This research was partially funded by the Italian Ministry of Education, Universities and Research, with the project PRIN 2015TW9LSR "Group theory and applications"

calculus algorithm consists of the following *relation collection* step and *linear algebra* step. In the *relation collection* step:

1. a factor base $\mathcal{F} \subset E(\mathbb{K})$ is defined;
2. for random integers u, v, the point $R = uP + vQ$ is computed;
3. if possible, R is written as a sum of multiples of points of $\mathcal{F}$:

$$R = uP + vQ = \sum_{F \in \mathcal{F}} \ell_F F, \tag{1}$$

   with the integers ℓ_F's ranging in a small coefficient set;
4. u, v and the vector (ℓ_F) are stored as a row of a matrix M;
5. the procedure from item 2 to item 4 is repeated until at least $|\mathcal{F}|$ points R as in (1) are found.

After the collection of a large enough number of relations, the *linear algebra* step solves the discrete logarithm problem, as follows:

1. by using linear algebra on M, a linear dependency of points R is computed, obtaining the relation $\lambda P + \mu Q = \infty$ with $\lambda, \mu \in \mathbb{Z}$;
2. w is recovered from the linear congruence $\lambda + \mu w = 0 \pmod{r}$, which is solvable except in the extremely unlikely case when $\mu = 0$.

The complexity of an index calculus algorithm mainly lies in the decomposition of a point R as sum of multiples of points in $\mathcal{F}$, usually known as the *point decomposition problem* (PDP). The solution of the PDP must be efficient (including the decision on the decomposability of a point R) and with a high success rate. Both these features are deeply affected by how $\mathcal{F}$ is defined and by its size.

In the third section of Semaev's paper [20], he proposes to reduce the PDP to the problem of finding specific solutions of a multivariate polynomial equation deduced from summation polynomials. Semaev sketched his proposal in the case of elliptic curves defined over prime fields $\mathbb{F}_p$, suggesting to define $\mathcal{F}$ as the set of rational points whose x-coordinates are *small* when taken as integers in the standard complete residue system $[0, \ldots, p-1]$. Furthermore, in Remark 2 of [20], Semaev also outlines a variation of his proposal to binary elliptic curves defined over extension fields. This variation involves the Weil descent, which cannot be applied to prime fields.

Semaev's paper triggered a deep interest in summation polynomials. Starting from them, Diem claimed that the ECDLP for some elliptic curves could be solved with a subexponential-time complexity (rather than exponential) in his 2006 talk [1]. However, the first papers developing Semaev's approach appeared only in 2009, by Gaudry [8], and then in 2011 by Diem [2]. In [8] Gaudry shows that Semaev's proposal can be used to solve, in heuristic time $\tilde{\mathcal{O}}(q^{2-\frac{2}{n}})$, the ECDLP for elliptic curves defined over finite fields of size q^n (from now on, ECDLP(q,n)), with q prime or a prime power and $n \geq 2$. However, Gaudry's results cannot be extended to the prime-field case. Diem [2] proved that the same problem can be solved in an expected time of $e^{\mathcal{O}(max(\log q, n^2))}$. Unfortunately, when $q = 2$ and n is large (one of the two common cases in leading cryptographic standards), the above works do not lead to an algorithm more effective than Pollard's Rho algorithm. In 2012 Faugère, Perret, Petit and Renault [5] claimed that, under a heuristic assumption, the ECDLP over any binary field $\mathbb{F}_{2^n}$ can be solved in time $\mathcal{O}(2^{\omega t})$, with $t \approx n/2$ and where $2.376 \leq \omega \leq 3$ is the *linear algebra constant*. Petit and Quisquater [18] revisited polynomial systems proposed by Faugère *et al.* in [5], introducing a heuristic assumption, named *first fall degree assumption*, under which the ECDLP(2,n) can be solved in time $\mathcal{O}(2^{cn^{2/3}\log n})$, where c is a constant smaller than 2.
Further relevant results were then accomplished by exploiting symmetries in the works of Huang, Petit, Shinohara and Takagi [9], of Faugère, Gaudry, Huot and Renault [3], and of Galbraith and Gebregiyorgis [7]. In the latter paper, the authors propose to use disjoint factor bases $\mathcal{F}_i$ and to require $P_i \in \mathcal{F}_i$ in (1).
Huang *et al.* [10], Semaev [21] and Karabina [12] had independently the idea to lower the degrees of polynomial equation systems, appearing in the PDP, at the cost of a larger number of variables. Experimental

results reported in that three works suggest this approach can lead to smaller running times.

Assuming the *first fall degree assumption* (which is not universally accepted), the listed result suggest a huge improvement over the index calculus algorithms for elliptic curves. However, the question whether the index calculus algorithm proposed by Semaev in 2004 can lead to a subexponential algorithm for binary elliptic curves is nowadays still open, since the methods have only be applied to very small parameters.

Even if the original proposal in Semaev's paper directly considered the case of elliptic curves defined over prime fields, only one recent paper ([17], 2016) takes such demanding case into account. Among the main obstacles that one may encounter when facing this case, it is worth mentioning the difficulty to endow $\mathcal{F}$ with an algebraic structure, and the impossibility to exploit the Weil descent in the PDP. Nevertheless, elliptic curves defined over prime fields are nowadays widely spread in cryptographic applications, as for example the Bitcoin system [16]. In [17] Petit *et al.* deal with the identification of a suitable $\mathcal{F}$ by introducing rational (or algebraic) maps L, obtained by a preprocessing specific for a given curve E, and defining $\mathcal{F}$ as the set of rational points $\{(x, y) \in E(\mathbb{F}_p) | L(x) = 0\}$, where p is a prime integer. In particular, the maps L can be decomposed as low degree maps, increasing the efficiency of their approach.

In this paper we propose a new variant of index calculus algorithms for elliptic curves that holds for any finite field, but which is particularly suitable for the prime-field case. At the beginning of our algorithm (item 1 of the *relation collection* step), we define the size of $\mathcal{F}$ but not $\mathcal{F}$ itself, which instead will be defined *on the run*. Indeed, every point R produced in item 2 of the *relation collection* step is appended to $\mathcal{F}$, giving a trivial decomposition of R in item 3. We proceed in this fashion until $\mathcal{F}$ contains the predetermined size. At this point a linear dependency among points in $\mathcal{F}$ is obtained by solving a point decomposition problem with summation polynomials. Consequently, our proposal reduces the index calculus algorithm for elliptic curves to the computation of a single linear relation among points of the factor base.
Our approach determines an evident advancement in the prime-field case. Concerning such a case, we compare the complexity of our algorithm with that by Petit *et al.*, showing that we improve the complexity of the algorithm in [17]. Nevertheless, more work is needed in order to reach a possible enhancement on Pollard's Rho algorithm for the prime-field case. We hope that our proposal can encourage further discussions on the prime-field case, which has been somewhat neglected so far.

The paper is organised as follows. In Section 2 we provide some background on summation polynomials and the relevant index calculus algorithm for elliptic curves. In Section 3 we introduce our index calculus algorithm, providing some complexity evaluations and experimental results for the prime-field case. Section 4 concludes the paper with comments and future perspectives.

2. Summation polynomials and the index calculus algorithm for elliptic curves

In 2004 Semaev introduced summation polynomials in order to use them to build an index calculus algorithm for elliptic curves. Let $\mathbb{K}$ be a field of any characteristic. Given an elliptic curve E defined over $\mathbb{K}$, for each $m \geq 2$ the *m-th* summation polynomial S_m is defined as follows:

Definition 1. *Let $\overline{\mathbb{K}}$ be the algebraic closure of the field $\mathbb{K}$. For any integer $m \geq 2$, the m-th summation polynomial S_m is an element of $\mathbb{K}[X_1, \ldots, X_m]$ and it is such that, given $x_1, \ldots, x_m \in \overline{\mathbb{K}}$, then $S_m(x_1, \ldots, x_m) = 0$ if and only if there exist $y_1, \ldots, y_m \in \overline{\mathbb{K}}$ for which $(x_1, y_1), \ldots, (x_m, y_m) \in E(\overline{\mathbb{K}})$ and*

$$(x_1, y_1) + \ldots + (x_m, y_m) = \infty.$$

The existence of summation polynomials is proved by providing recursive formulae, along the following properties.

Theorem 2 ([20]). *Let E be an elliptic curve defined over a field $\mathbb{K}$, with char$(\mathbb{K})$ different from 2 and 3, by the Weierstrass equation*

$$y^2 = x^3 + Ax + B.$$

The summation polynomials for E are given as follows:

$$\begin{aligned} S_2(X_1, X_2) &= X_1 - X_2, \\ S_3(X_1, X_2, X_3) &= (X_1 - X_2)^2 X_3^2 - 2[(X_1 + X_2)(X_1X_2 + A) + 2B]X_3 + \\ &\quad + (X_1X_2 - A)^2 - 4B(X_1 + X_2), \end{aligned}$$

and, for all $m \geq 4$ and $1 \leq k \leq m-3$, it holds

$$S_m(X_1, \ldots, X_m) = \mathrm{Res}_X(S_{m-k}(X_1, \ldots, X_{m-k-1}, X), S_{k+2}(X_{m-k}, \ldots, X_m, X))$$

where Res_X denotes the resultant polynomial of S_{m-k} and S_{k+2} with respect to the variable X. Moreover, for any $m \geq 3$, the polynomial S_m is symmetric of degree 2^{m-2} in each variable X_i, it is absolutely irreducible and

$$S_m(X_1, \ldots, X_m) = S_{m-1}^2(X_1, \ldots, X_{m-1})X_m^{2^{m-2}} + I_m$$

where I_m is a polynomial of $\mathbb{K}[X_1, \ldots, X_m]$ such that $\deg_{X_m}(I_m) < 2^{m-2}$.

In the case of elliptic curves defined over fields of characteristics 2 or 3, it turns out that summation polynomials exist and have a similar form (and the same properties, see [20], [2]).

2.1. Index calculus algorithm for elliptic curves

In [20], Semaev proposes to use summation polynomials for an index calculus algorithm solving the discrete logarithm problem for elliptic curves. Semaev's proposal was firstly developed by Gaudry [8] and Diem [2], and then many other papers on this topic followed. We describe here the current approach to these ideas. As we pointed out previously, all works subsequent to [20], except one, take into account only elliptic curves defined over extension fields.

Consider an elliptic curve E defined over a finite field $\mathbb{F}_{q^n}$, where q is a prime or a prime power and $n > 1$. Let $P \in E(\mathbb{F}_{q^n}) \setminus \{\infty\}$ and $Q \in < P >$, where $< P >$ denotes the cyclic group generated by P. It is standard to restrict attention to points P of prime order r. Concerning the *relation collection* step, the general procedure is the following:

1. An $\mathbb{F}_q$-vector subspace V of $\mathbb{F}_{q^n}$ is fixed, with $|V| = q^\ell$ where $1 \leq \ell < n$. Then the factor base $\mathcal{F}$ is defined as:

$$\mathcal{F} = \{F \in E(\mathbb{F}_{q^n}) \mid x(F) \in V\} \tag{2}$$

where $x(F)$ denotes the x-coordinate of F.
2. Two random integers u, v are generated and the point $R = uP + vQ$ is computed.
3. For a fixed integer m, points $P_i \in \mathcal{F}$ with $i \in \{1, \ldots, m\}$ and such that

$$R = P_1 + \cdots + P_m \tag{3}$$

are searched. Such P_i's are obtained by solving the polynomial equation

$$S_{m+1}(X_1, \ldots, X_m, x(R)) = 0 \tag{4}$$

and considering solutions of the form $(x_1, \ldots, x_m) \in (\tilde{V})^m$, where $\tilde{V}$ is the biggest subset of V such that for every $x \in \tilde{V}$, there exists a rational point of E with x as abscissa. Hence the point decomposition problem is reduced to the solution of a multivariate polynomial equation thanks to summation polynomials. If R is

not decomposable as a sum of points of $\mathcal{F}$, it is replaced by a new R coming back to step 2.
4. To every $x \in \tilde{V}$, one point in $\mathcal{F}$ having x as abscissa is chosen. Consequently, R can be written as $R = \pm P_1 \pm \cdots \pm P_m$, where $P_i \in \mathcal{F}$ and with signs depending on the selected points. A matrix M with $|\tilde{V}| + 2$ columns is then constructed: each of the first $|\tilde{V}|$ columns corresponds to a single element of $\tilde{V}$, while the two last columns correspond to the coefficients u and v, respectively. Given the relation $R = \pm P_1 \pm \cdots \pm P_m$, a row of M is filled with the coefficients of the chosen points and the coefficients u, v such that $R = uP + vQ$.
5. The three steps above are repeated until at least $|\tilde{V}|$ relations have been computed, i.e. at least $|\tilde{V}|$ rows of M have been filled.

At this point the integer w such that $Q = wP$ can be computed by executing the *linear algebra* step:

1. via Gauss' elimination, a linear dependency among some M's rows is obtained, and then its corresponding linear dependency of points R. This leads to the sought-after relation

$$\lambda P + \mu Q = \infty \tag{5}$$

with $\lambda, \mu \in \mathbb{Z}$.
2. w is recovered from the modular equation $\lambda + \mu w = 0 \pmod{r}$ if it is solvable.

As observed, while considering elliptic curves defined over extension fields, the subset $V \subset \mathbb{F}_{q^n}$ is usually chosen as an $\mathbb{F}_q$-vector subspace of $\mathbb{F}_{q^n}$. Consequently, the Weil descent can be exploited to replace the polynomial equation system in the PDP with an equivalent one, which have polynomial equations over $\mathbb{F}_q$. Systems that arise are then usually solved with Groebner basis methods ([5], [18], [11], [22], [3]).

2.2. The prime-field case

The case when the considered elliptic curve E is defined over a prime field $\mathbb{F}_p$ was originally taken into account by Semaev in [20], where he defines V as the subset $\{x \in \mathbb{F}_p \mid x < p^{1/m+\delta}\}$ for a fixed m and some small $\delta > 0$. However, Semaev did not provide an algorithm to solve the systems arising in the point decomposition problems and this case was somewhat neglected so far.
In 2016, Petit *et al.* were the first, after Semaev, to face the prime-field case. In [17], they primarily observe that the constraints $x_i \in \tilde{V}$ in item 3 of the above *relation collection* step are equivalent to the constraints $L(X_i) = 0$ where $L(z)$ is a polynomial defined as:

$$L = \prod_{v \in \tilde{V}} (z - v)$$

This observation can be exploited to transform Semaev's prime-field proposal into an actual algorithm. Furthermore, Petit *et al.* generalised it considering *suitable* algebraic (or rational) maps L over $\mathbb{F}_p$ - i.e. maps L that are composition of low degrees maps $L_1, \ldots, L_t$ - and defining V as the set $\{x \in \mathbb{F}_p \mid L(x) = 0\}$. Accordingly, the factor base $\mathcal{F}$ is then defined as $\{F \in E(\mathbb{F}_p) \mid x(F) \in V\}$, where $x(F)$ is the abscissa of F. To obtain a suitable map L, they introduced a pre-computation, which is specific for every elliptic curve E.

The index calculus algorithm considered by Petit *et al.* to solve the ECDLP in the prime-field case is the same described in the previous section. However, given a point $R = uP + vQ \in E(\mathbb{F}_p)$, in order to solve the point decomposition problem for R with respect to $\mathcal{F}$, they suggested to solve the following polynomial equation system:

$$\begin{cases} S_{m+1}(X_{11}, \ldots, X_{m1}, x(R)) = 0 & \\ X_{i,j+1} = L_j(X_{i,j}) & i = 1, \ldots, m; \quad j = 1, \ldots, t-1 \\ L_t(X_{i,t}) = 0 & i = 1, \ldots, m \end{cases} \tag{6}$$

where $x(R)$ is the x-coordinate of R. In their work, Petit *et al.* provide some experimental results and partial complexity analysis, with which we are going to compare.

3. Our proposal

In this section we introduce a new variant of the index calculus algorithm for elliptic curves based on summation polynomials. Our proposal differs significantly from the algorithm reported in Section 1, since after the *relation collection* step we do not have a *linear algebra* step, but rather a *relation solving* step, as we detail in the following.

Let us consider an elliptic curve E defined over a finite field $\mathbb{F}_q$ (where q is a prime or a prime-power) and having a prime number of rational points, i.e. $|E(\mathbb{F}_q)| = r$ and r prime integer. With P and Q we denote two points of $E(\mathbb{F}_q)$ such that $Q = wP$.

Relation collection step
1. The factor base $\mathcal{F}$ is not fixed, rather it is initialized as the empty set. The final size s of $\mathcal{F}$ is fixed together with an integer m, named *decomposition constant*. As usual (see for example [17, pag.7]) we take $s^m \approx q$.
2. For random integers u, v, the point $R = uP + vQ$ is computed and added to the factor base $\mathcal{F}$.
3. R does not need to be written as a sum of multiples of points in $\mathcal{F}$, since R itself is a point of $\mathcal{F}$.
4. The procedure in item 2 is repeated until at least s different points are added to $\mathcal{F}$.

Relation solving step
1. A linear dependency among points in $\mathcal{F}$ is determined using summation polynomials instead of linear algebra. To be more precise, we consider the multivariate polynomial equation $S_m(X_1, \ldots, X_m) = 0$, where S_m is the *m-th* summation polynomial for E, and we search solutions of the form $(x_1, \ldots, x_m) \in V^m$ where:

$$V = \{x \mid x = x(R) \text{ for some } R \in \mathcal{F}\}. \tag{7}$$

If the equation has not suitable solutions, then we come back to the relation collection step, in order to create a new factor base $\mathcal{F}$. Otherwise, if an appropriate solution has been found, it is trivial to determine m points P_i such that $-P_i \in \mathcal{F}$ or $P_i \in \mathcal{F}$, and

$$P_1 + \cdots + P_m = \infty. \tag{8}$$

Hence, the linear congruence $\lambda P + \mu Q = \infty$, with $\lambda, \mu \in \mathbb{Z}$, is deduced.
2. The discrete logarithm w of Q with respect to P is recovered solving the modular equation $\lambda + \mu w = 0 \pmod{r}$.

The algorithm described above differs in three key points from the standard index calculus algorithm usually considered when exploiting summation polynomials. First, $\mathcal{F}$ is not fixed deterministically at the beginning of the algorithm. Instead, it is constructed step by step adding the random points R progressively. This prevents the computation of many relations (i.e. resolutions of the PDP), which traditionally requires expensive Grobner basis computations. Second, every element in V corresponds to 2 points on the curve, whereas in previous methods it is heuristically expected that half of the element in V correspond to 2 rational points. Third, in our second step (which is traditionally the *linear algebra* step) we actually do not execute linear algebra. On the other hand, the linear dependency among points R is obtained by looking for suitable solutions of the equation $S_m(X_1, \ldots, X_m) = 0$. To do that, we need to compute a Groebner basis. If the considered equation has not a suitable solution, then we change the factor base $\mathcal{F}$ and we solve a new equation. Usually more than one attempt should be made to find the single relation among points of the factor base $\mathcal{F}$, but the number of computed Groebner bases is decreased dramatically.
We now specify which is the ideal basis that we use for our Groebner basis computation.

3.1. The system to be solved

In order to deduce a linear relation among points of the factor base $\mathcal{F}$, we solve the polynomial equation $S_m(X_1, \ldots, X_m) = 0$ considering only solutions of the form $(x_1, \ldots, x_m) \in V^m$. Formally, this is reached

solving the polynomial equation system

$$\begin{cases} S_m(X_1, \ldots, X_m) = 0 \\ f(X_1) = 0 \\ \ldots \\ f(X_m) = 0 \end{cases} \tag{9}$$

where $f(z)$ is the polynomial generating the vanishing ideal of $V \subset \mathbb{F}_q$. In particular:

$$f(z) = \prod_{v \in V} (z - v)$$

Although we would be content with any solution of $S_m(X_1, \ldots, X_m) = 0$ of the form

$$(x_1, \ldots, x_m) \ \in \ V^m,$$

we add further constraints on the corresponding variables $X_1, \ldots, X_m$ in order to lower the degrees in system (9). Indeed, given any partition $\{\mathcal{F}_i \mid i = 1, \ldots, m\}$ of $\mathcal{F}$, the sets

$$V_i = \{x \mid x = x(R) \text{ for some } R \in \mathcal{F}_i\} \qquad i = 1, \ldots, m$$

can be trivially deduced and the following polynomials constructed:

$$f_i(X_i) = \prod_{v \in V_i} (X_i - v) \qquad i = 1, \ldots, m. \tag{10}$$

The idea of partitioning $\mathcal{F}$ is similar to what Galbraith and Gebregiyorgis propose in [7] and it allows us to limit the degrees of the univariate polynomials in $X_1, \ldots, X_m$ respectively. Therefore, the system to be solved in item 1 of the *Relation solving* step of our algorithm is the following:

$$\begin{cases} S_m(X_1, \ldots, X_m) = 0 \\ f_1(X_1) = 0 \\ \ldots \\ f_m(X_m) = 0. \end{cases} \tag{11}$$

The constraints on $X_1, \ldots, X_m$ in system (11) break the symmetry and avoid trivial solutions when m is even, since m disjoint factor bases are used and only $P_i \in \mathcal{F}_i$ (or $-P_i \in \mathcal{F}_i$) are then allowed in (8). This will obviously impact the probability of finding a solution of system (11): a complexity analysis is reported in the next subsection.

3.2. Complexity analysis

The whole complexity of the proposed algorithm relies on the solution of several polynomial equation systems. Let $T(E, m, V)$ be the computational cost of solving system (11). Under the assumption that the sizes of the $\mathcal{F}_i$'s are about the same (which is in practice what we always enforce), the expected number of linear combinations $P_1 + \cdots + P_m$ (with P_i or $-P_i$ in $\mathcal{F}_i$) is about

$$\left(\frac{s}{m}\right)^m .$$

Then the probability that a random point of $E(\mathbb{F}_q)$ could be written as a sum $P_1 + \cdots + P_m$ (with P_i or $-P_i$ in $\mathcal{F}_i$) is nearly

$$\frac{s^m}{q \ m^m} . \tag{12}$$

Therefore, the total cost of the algorithm is given by:

$$\frac{q\ m^m}{s^m} T(E, m, V).$$

In order to reach an improvement with respect to algorithms for generic cyclic groups, this complexity should be smaller than $r^{1/2}$. We do not claim this result. Indeed our complexity analysis is partial, since we are unable to estimate the complexity of solving our polynomial equation systems.

Our approach can be applied to any finite field. However, in the composite-field case it must face the improvement given by the use of the Weil descent when V is chosen as a vector subspace. Such a drawback does not hold for the prime-field case, where the enhancement of our algorithm is evident, as we will show in the next section comparing our complexity with that of the algorithm of Petit *et al.*.

3.3. Complexity analysis for the prime-field case

Following the notation of our paper and considering an elliptic curve defined over a prime field $\mathbb{F}_p$, the complexity of the algorithm of Petit *et al.* can be expressed as:

$$\frac{p\ m!}{s^{m-1}} T'(E, m, V) + s^\omega$$

where:

- $T'(E, m, V)$ is the cost to solve one of the systems that arise executing the algorithm proposed in [17], which are of the form of system (6);
- s^ω is the linear algebra cost, with $2 < \omega \leq 3$.

Neglecting the linear algebra cost and assuming

$$T(E, m, V) \ \approx \ T'(E, m, V), \tag{13}$$

where $T(E, m, V)$ is the complexity of solving the system (11) for the elliptic curve E, for a fixed m our algorithm outperforms that of Petit *et al.* when p increases. In particular, the outperforming happens when

$$s(m!) > m^m\,, \qquad \text{that is,} \qquad \sqrt[m]{p} > \frac{m^m}{m!}.$$

Considering $m \in \{3, 4, \ldots, 8\}$ (S_8 is nowadays the summation polynomial with the highest index that has been computed [4]), the following table shows from which bit size of p our algorithm would outrun that in [17].

m	3	4	5	6	7	8
bit size	7	14	24	37	52	70

Table 1: Bit sizes of p from which our algorithm improves Petit *et al.* algorithm.

We underline that the bit sizes of p reported in the above table are all smaller than the bit sizes used in cryptographic applications.

The systems arising in our algorithm need to be analysed more deeply in order to reach a more precise complexity evaluation and it should be also inspected whether algorithms different from Grobner basis methods are more suitable to tackle such systems. Hence, proving theoretically the assumption (13) is a challenging issue. However, the experimental results reported in next section show that the assumption has heuristic evidence.

3.4. Experimental results for the prime-field case

We report some computer experiments, executed on elliptic curves defined over prime fields $\mathbb{F}_p$ with the bit sizes of p lying in $\{11, 12, \ldots, 22\}$. The possible values for the *decomposition constant* m that we considered belong to the set $\{3, 4, 5\}$. Following [10], [12] and [21], for $m = 4$ we substituted $S_4(X_1, X_2, X_3, X_4) = 0$ with

$$\begin{cases} S_3(X_1, X_2, Y) = 0 \\ S_3(X_3, X_4, Y) = 0 \end{cases}$$

introducing the variable Y and obtaining the following system in place of (11):

$$\begin{cases} S_3(X_1, X_2, Y) = 0 \\ S_3(X_3, X_4, Y) = 0 \\ f_1(X_1) = \cdots = f_4(X_4) = 0. \end{cases}$$

Similarly, for $m = 5$ the variable Y was introduced, obtaining the following system in place of (11):

$$\begin{cases} S_3(X_1, X_2, Y) = 0 \\ S_4(X_3, X_4, X_5, Y) = 0 \\ f_1(X_1) = \cdots = f_5(X_5) = 0. \end{cases}$$

For every possible values of m and the bit size of p, we considered 100 different elliptic curves. For each of them, we use our algorithm to solve 10 instances of the discrete logarithm problem, taking the average execution time and the average size of the factor bases $\mathcal{F}$. Executing our algorithm, several polynomial equation systems should be solved, since the probability to find a suitable solution satisfying a single system (11) is reported in equation (12). Of the 100 average times and the 100 average sizes obtained (one for every elliptic curve), we considered the averages (in the following table, "Av. time" and "Av. size" respectively). All the experiments were performed with MAGMA on a CPU with an Intel Xeon Process 5460 at 3.16 GHz with a cache of 6 MB. The collected data are showed in the following table, together with an extra column with the experimental results reported in [17] and concerning the average time necessary to solve one of the systems arising in Petit *et al.* algorithm. Their data for even bit sizes are relatively to primes of a specific form; the ones for odd bit sizes are relatively to generic primes. All of them were obtained exploiting a computational power similar to ours.

bit	$m = 3$	$m = 3$		$m = 4$		$m = 5$	
size	Petit *et al.*	Av. time	Av. size	Av. time	Av. size	Av. time	Av. size
11	0.02	0.007	13	0.010	7	0.054	6
12	0.15	0.016	15	0.021	8	0.085	6
13	0.13	0.036	20	0.042	10	0.121	7
14	1.16	0.081	24	0.089	12	0.136	8
15	1.14	0.228	30	0.214	14	0.201	8
16	9.08	0.506	38	0.425	16	0.334	10
17	9.09	1.377	47	0.950	19	0.906	11
18	51.87	3.303	59	2.755	22	2.675	13
19	59.87	10.120	74	8.175	26	10.585	14
20	438.57	28.230	92	31.593	30	35.635	16
21	454.79	99.860	116	103.193	37	120.575	19
22	5163.46	267.659	146	339.906	43	387.981	21

Table 2: Average times and sizes of $\mathcal{F}$ obtained running our algorithm for different primes and different decomposition constants. The second column reports from [17] the average times of Petit *et al.* algorithm for $m = 3$.

Considering the case $m = 3$, it is important to notice that the results of Petit *et al.* refer to the average time necessary to solve one of the systems that need to be tackled in the execution of their algorithm, while our results refer to the average time necessary for solving an instance of the ECDLP, which requires the resolution of several systems. So the times reported in the second column should be multiplied by a factor bigger than s to obtain the time needed by the Petit *et al.* algorithm to solve an instance of the ECDLP. For example, if p has a bit size equal to 11, such a factor is bigger than 10. The same remark holds considering Semaev's original proposal in [20]. Then, a comparison of the reported times heuristically corroborates the assumption (13) that we considered in the complexity analysis for the prime-field case.

We performed many preliminary computations with several monomial orderings available in MAGMA. Since we did not find any which would clearly outperforms the others with our systems, we decided to use only the "graded reverse lexicographic order" for the computations reported in the table.

4. Conclusions

We have presented a new index calculus algorithm that exploits summation polynomials for solving the discrete logarithm problem on elliptic curves defined over finite fields. This algorithm significantly differs from the other index calculus algorithms and it reduces to one the number of relations to be found among points of the factor base. Our algorithm can be applied to any finite field but its improvement is evident for the prime-field case. Such a case was somewhat neglected so far, with the exception of a recent paper by Petit *et al.* A preliminary complexity analysis shows that our algorithm improves the one of Petit *et al.* when m is fixed and p increases.

The complexity of the systems to be tackled in our algorithm is far to be well understood and it needs to be deeply analysed. It should be also investigated whether other algorithms different from Grobner basis algorithms are better to solve such systems. Furthermore additional work should be executed in order to determine if our approach can lead to an improvement also in the composite-field case.

References

[1] C. Diem, On the Subexponentiality of the Elliptic Curve Discrete Logarithm Problem over Extension Fields, Workshop "Grobner Bases in Cryptography, Coding Theory, and Algebraic Combinatorics" organised by Mikhail Klin, Ludovic Perret, Massimiliano Sala, May 2006.

[2] C. Diem, On the discrete logarithm problem in elliptic curves, Compositio Mathematica, 147 (2011) 75-104.

[3] J. C. Faugère, P. Gaudry, L. Huot, G. Renault, Using symmetries in the index calculus for elliptic curves discrete logarithm, Journal of Cryptology, 27(4) (2014) 595-635.

[4] J. C. Faugère, L. Huot, A. Joux, G. Renault, V. Vitse, Symmetrized summation polynomials: using small order torsion points to speed up elliptic curve index calculus, Annual International Conference on the Theory and Applications of Cryptographic Techniques (Springer Berlin Heidelberg), (2014) 40-57.

[5] J. C. Faugère, L. Perret, C. Petit, G. Renault, Improving the complexity of index calculus algorithms in elliptic curves over binary fields, Annual International Conference on the Theory and Applications of Cryptographic Techniques (Springer Berlin Heidelberg), 2012.

[6] S. D. Galbraith, P. Gaudry, Recent progress on the elliptic curve discrete logarithm problem, Designs, Codes and Cryptography, 78.1 (2016) 51-72.

[7] S. D. Galbraith, S. W. Gebregiyorgis, Summation polynomial algorithms for elliptic curves in characteristic two, International Conference in Cryptology in India (Springer International Publishing), (2014) 409-427.

[8] P. Gaudry, Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem, Journal of Symbolic Computation, 44(12) (2009) 1690-1702.

[9] Y. J. Huang, C. Petit, N. Shinohara, T. Takagi, Improvement of Faugère *et al.*'s Method to Solve ECDLP, International Workshop on Security (Springer Berlin Heidelberg), (2013) 115-132.

[10] Y. J. Huang, C. Petit, N. Shinohara, T. Takagi, On Generalized First Fall Degree Assumptions, IACR Cryptology ePrint Archive, 2015/358.

[11] A. Joux, V. Vitse, Elliptic curve discrete logarithm problem over small degree extension fields, Journal of Cryptology, (2013) 1-25.

[12] K. Karabina, Point decomposition problem in binary elliptic curves, International Conference on Information Security and Cryptology (Springer International Publishing), 2015.

[13] N. Koblitz, Elliptic curves cryptosystems, Mathematics of Computation, 48 (177) (1987) 203-209.

[14] A. Menezes, S. Vanstone, T. Okamoto, Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field, iEEE Transactions on information Theory, 39.5 (1993) 1639-1646.
[15] V. S. Miller, Use of elliptic curves in cryptography, Conference on the Theory and Application of Cryptographic Techniques (Springer Berlin Heidelberg), 1985.
[16] Satoshi Nakamoto, Bitcoin: A peer-to-peer electronic cash system, https://bitcoin.org/bitcoin.pdf, 2008.
[17] C. Petit, M. Kosters, A. Messeng, Algebraic Approaches for the Elliptic Curve Discrete Logarithm Problem over Prime Fields, IACR International Workshop on Public Key Cryptography (Springer Berlin Heidelberg), (2016) 3-18.
[18] C. Petit, J. J. Quisquater, On polynomial systems arising from a Weil descent, International Conference on the Theory and Application of Cryptology and Information Security (Springer), (2012) 451-466.
[19] J. M. Pollard, Monte Carlo methods for index computation mod p, Mathematics of computation, 32.143 (1978) 918-924.
[20] I. Semaev, Summation polynomials and the discrete logarithm problem on elliptic curves, IACR Cryptology ePrint Archive, 2004/31.
[21] I. Semaev, New algorithm for the discrete logarithm problem on elliptic curves, arXiv:1504.01175, 2015.
[22] M. Shantz, E. Teske, Solving the elliptic curve discrete logarithm problem using Semaev polynomials, Weil descent and Groebner basis methods. An experimental study, Number Theory and Cryptography (Springer), (2013) 94-107.