Algebraic approaches for the Elliptic Curve Discrete Logarithm Problem over prime fields

Christophe Petit[1], Michiel Kosters[2], and Ange Messeng[3]

[1] Mathematical Institute, University of Oxford
Andrew Wiles Building
Radcliffe Observatory Quarter
Woodstock Road
Oxford OX2 6GG
United Kingdom
christophe.f.petit@gmail.com

[2] University of California, Irvine
340 Rowland Hall
Irvine, CA 92697-3875
United States of America
kosters@gmail.com

[3] Faculty of Mathematics, University of Passau
InnStrasse 33, IM207
94032 Passau
Germany
messeng.ange2@gmail.com

Abstract. The elliptic curve discrete logarithm problem is one of the most important problems in cryptography. In recent years, several index calculus algorithms have been introduced for elliptic curves defined over extension fields, but the most important curves in practice, defined over prime fields, have so far appeared immune to these attacks.
In this paper we formally generalize previous attacks from binary curves to prime curves. We study the efficiency of our algorithms with computer experiments and we discuss their current and potential impact on elliptic curve standards.
Our algorithms are only practical for small parameters at the moment and their asymptotic analysis is limited by our understanding of Gröbner basis algorithms. Nevertheless, they highlight a potential vulnerability on prime curves which our community needs to explore further.

1 Introduction

The elliptic curve discrete logarithm problem (ECDLP) is widely believed to be one of the hardest computational number theory problem used in cryptography. While integer factorization and discrete logarithms over finite fields suffer from index calculus attacks of subexponential or even quasipolynomial complexity,

recommended key sizes for elliptic curve cryptography correspond to the birthday paradox bound complexity of generic discrete logarithm algorithms.

In the last ten years starting from the seminal work of Semaev [20], index calculus algorithms have progressively been adapted to elliptic curve discrete logarithm problems. However, the most efficient attacks target parameters that are not used in standards; attacks against binary curves rely on poorly understood Gröbner basis assumptions; and almost no attacks at all have been proposed against the most important family of curves, namely elliptic curves defined over prime fields.

Contributions In this paper, we provide new index calculus algorithms to solve elliptic curve discrete logarithm problems over prime fields of cardinality p.

The factor bases in our algorithms are of the form $\mathcal{F} := \{(x, y) \in E(K) | L(x) = 0\}$, where L is a large-degree rational map. We additionally require that L is a composition of small-degree rational maps L_j, $j = 1, \ldots, n'$, such that the large-degree constraint $L(x) = 0$ can be replaced by a system of low degree constraints $x_2 = L_1(x)$, $x_3 = L_2(x_2)$, $x_4 = L_3(x_3)$, $\ldots$, $x_{n'} = L_{n'-1}(x_{n'-1})$, $L_{n'}(x_{n'}) = 0$. Relations are computed by solving a polynomial system constructed from Semaev's summation polynomials and the above decomposition of the map L.

Our factor bases generalize the factor bases used in previous works: Diem and Gaudry's attacks [11,7] implicitly use $L(x) = x^q - x$ where q is the size of a subfield; small characteristic, prime degree extension attacks [8,9,18,22,13,10] implicitly use the linearized polynomial corresponding to a vector space; and Semaev's original factor basis [20] implicitly uses $L(x) = \prod_{\alpha<B}(x - \alpha)$ for some B of appropriate size. The potential advantage of our polynomials L compared to the one implicitly used by Semaev is that they can be re-written in the form of a system of low degree polynomial equations, similar to systems occurring in the characteristic 2 case, which we then solve using Gröbner basis algorithms.

We specify two concrete instances of the above algorithm. In the first instance, we assume that $p - 1$ has a large divisor which is smooth, and we define L such that its roots form precisely a coset of a subgroup of smooth order. In the second instance, we assume the knowledge of an auxiliary curve over the same field with a large enough smooth subgroup, and we define L using the isogeny corresponding to that subgroup. We complete the second instance with two different algorithms to compute an auxiliary curve over a finite field, and we compare both methods.

Interestingly, the standardized curve NIST P-224 falls into the framework of our first algorithm. We also show that computing a finite field and an auxiliary curve for this field is as far as we know much easier than computing an auxiliary curve for a given finite field.

The complexity of our algorithms remains an open problem. We implemented both of them in Magma, and compared their performances to previous attacks on binary curves of comparable sizes. The experimental results suggest that in spite of a common structure, the systems are a bit more efficient to solve in binary cases than in prime cases. They also suggest that all the systems we studied are easier to solve than generic systems of "comparable parameters". This may look

encouraging from a cryptanalytic point of view, but we stress that the set of experiments is too limited to draw any conclusion at this stage (see also [12] for a criticism of the analysis of [18]). At the moment all attacks are outperformed by generic discrete logarithm algorithms for practically relevant parameters.

Perspectives Our paper introduces a new algorithmic framework to solve ECDLP over prime fields. We hope that these ideas revive research in this area and lead to a better understanding of the elliptic curve discrete logarithm problem.

Proving meaningful complexity bounds for our algorithms appears very challenging today as they use Gröbner basis algorithms on non-understood families of polynomial systems with a special structure. At the time of writing it is not clear yet whether the special structure introduced in this paper leads to asymptotic improvements with respect to generic discrete logarithm algorithms. Of course, Gröbner basis algorithms may also not be the best tools to solve these systems. At the end of the paper we suggest that better, dedicated algorithms to solve these systems, inspired from existing root-finding algorithms, could perhaps lead to substantial efficiency and analysis improvements of our algorithms.

Related work In recent years many index calculus algorithms have been proposed for elliptic curves [20,11,7,8,9,18,22,13,10]. All these papers except Semaev's paper [20] focus on elliptic curves defined over extension fields, and Semaev did not provide an algorithm to compute relations. Moreover, our work offers a natural large prime counterpart to recent characteristic 2 approaches, and an avenue to generalize any future result on these approaches to the even more interesting large prime case.

We are aware of two other types of attacks that first exploited smoothness properties of $p-1$ and were later generalized using elliptic curves. The first one is Pollard's $p-1$ factorization method generalized to the celebrated elliptic curve factorization method [14]. The second one is den Boer's reduction of the computational Diffie-Helman problem to the discrete logarithm problem, which was generalized by Maurer [6,15]. We point out that the smoothness requirements on the auxiliary curve order are much weaker in our attacks than in these contexts.

Because of these attacks, there may also be a folklore suspicion in the community that using primes with special properties could lead to improved attacks on elliptic curves, but to the best of our knowledge this was not supported by any concrete attack so far, and in fact all NIST curves use generalized Mersenne primes.

Outline The remaining of the paper is organized as follows. In Section 2 we describe related work, particularly on binary curves. In Section 3 we describe our main results. We first sketch our main idea and provide a partial analysis of our general algorithm, leaving aside precomputation details and the complexity of the Gröbner basis step. We then describe the $p-1$ smooth and isogeny versions of our algorithms, and we analyze the complexity of computing an auxiliary curve in the second case. In Section 4 we describe our experimental evaluation of the

attack. Finally, Section 5 summarizes our results and provides routes towards improvements.

2 Previous work on (binary) curves

Let K be a finite field; let E be an elliptic curve defined over K; and let $P, Q \in E(K)$ such that Q is in the subgroup $G \subset E(K)$ generated by P. The discrete logarithm problem is the problem of finding an integer k such that $Q = kP$. In the following we assume that the order r of G is prime, as it is usually the case in cryptographic applications.

2.1 Index calculus for elliptic curves

Index calculus algorithms use a subset $\mathcal{F} \subset G$ often called a *factor basis*. The simplest algorithms run in two stages. The first stage consists in collecting *relations* of the form

$$a_i P + b_i Q + \sum_{P_j \in \mathcal{F}} e_{ij} P_j = 0.$$

The second stage consists in performing linear algebra on these relations to deduce a relation of the form

$$aP + bQ = 0,$$

from which the discrete logarithm $k = -a/b \bmod r$ is easily deduced.

Since the seminal work of Semaev [20], index calculus algorithms for elliptic curves have used a basis of the form

$$\mathcal{F} := \{(x, y) \in E(K) | x \in V\}$$

where V is some subset of K. Relations are obtained by computing $R = (X, Y) = aP + bQ$ for random a and b, then solving a polynomial equation

$$S_{m+1}(x_1, \ldots, x_m, X) = 0$$

with the additional constraints that $x_i \in V$ for all i. Here S_ℓ is such that for $X_1, \ldots, X_\ell \in \overline{K}$ one has $S_\ell(X_1, \ldots, X_\ell) = 0$ if and only if there exist $P_i = (X_i, Y_i) \in E(\overline{K})$ with $P_1 + \ldots + P_\ell = 0$. The polynomials S_ℓ are called *summation polynomials*.

When $K = \mathbb{F}_p$, Semaev originally proposed to use $V = \{x \in \mathbb{Z}_{\geq 0} | x < p^{1/m}\}$. This was inspired by the factor bases used for discrete logarithms over finite fields. However, Semaev did not suggest any algorithm to compute relations with this factor basis.

2.2 Weil restriction on vector spaces

In the case of an extension field $K = \mathbb{F}_{q^n}$, developments of Semaev's ideas by Gaudry, Diem and Faugère-Perret-Petit-Renault [11,7,8,9] led to choosing V as a linear subspace of $\mathbb{F}_{q^n}/\mathbb{F}_q$ with dimension $n' \approx \lceil n/m \rceil$. In order to compute relations, we then proceed to a *Weil descent* or *Weil restriction* of the summation polynomial onto the vector space.

Concretely we fix a basis $\{v_1, \ldots, v_{n'}\}$ for V, we define mn' variables x_{ij} over $\mathbb{F}_q$, we substitute x_i by $\sum_j x_{ij} v_j$ in S_{m+1}, and by fixing a basis $\{\theta_1, \ldots, \theta_n\}$ of $\mathbb{F}_{q^n}/\mathbb{F}_q$ we see the resulting equation over $\mathbb{F}_{q^n}$ as a system of n polynomial equations over $\mathbb{F}_q$. Namely, we write

$$S_{m+1}\left(\sum_j x_{1j} v_j, \ldots, \sum_j x_{mj} v_j, X\right) = 0$$

in the form

$$\sum_k \theta_k f_k(x_{ij}) = 0$$

which implies that for all k we have

$$f_k(x_{ij}) = 0.$$

This polynomial system is then solved using generic methods such as resultants or Gröbner basis algorithms.

A particular case of this approach consists in taking $V := \mathbb{F}_q$. The resulting index calculus algorithm is more efficient than generic algorithms for fixed $n > 3$ and large enough q, and has subexponential time when q and n increase simultaneously in an appropriate manner [11,7].

Another particular case occurs when q is a very small constant (typically $q = 2$). In this case the efficiency of Gröbner basis algorithms is increased by adding the so-called *field equations* $x_{ij}^q - x_{ij} = 0$ to the system. Experimental results and a heuristic analysis led Petit and Quisquater to conjecture that the algorithm could also have subexponential time in that case [18].

2.3 Limits of previous works

From a practical point of view, the subexponential result in [7] is of little interest as elliptic curves that appear in leading cryptographic standards are defined either over prime fields or binary fields with a prime extension degree. Semaev's seminal paper [20] proposes one factor basis for the prime case, but as mentioned above, it does not provide any corresponding algorithm to compute relations.

Binary curves may be vulnerable to index calculus algorithms for large enough parameters, according to Petit and Quisquater's analysis and following works [18,22,13,10]. However, generic algorithms currently outperform these algorithms for the parameters used in practice, and the complexity estimates for larger parameters depend on the so-called *first fall degree assumption*. This assumption on

Gröbner basis algorithms holds in some cases including for HFE systems [16,12], but it is also known to be false in general. The systems occurring in binary ECDLP attacks are related to HFE systems, but at the time of writing it is not clear whether or not, or to which extent the assumption holds in their case. On the other hand, as the systems in play are clearly not generic, one should a priori be able to replace Gröbner basis by other, more dedicated tools.

2.4 Alternative systems

One idea in that direction is to completely avoid the Weil descent methodology. The vector space constraints $x_i \in V$ are equivalent to the constraints $L(x_i) = 0$ where

$$L(x) := \prod_{v \in V} (x - v).$$

It is easy to prove (see [2, Ch. 11]) that L is a linearized polynomial, in other words L can be written as

$$L(x) = \sum_{j=0}^{n'} c_j x^{q^j}$$

where $c_j \in \mathbb{F}_{q^n}$. Moreover (see also [17]) L can be written as a composition of degree q maps

$$L(x) = (x^q - \alpha_{n'} x) \circ \ldots \circ (x^q - \alpha_1 x) \tag{1}$$

for well-chosen $\alpha_i \in \mathbb{F}_{q^n}$. Abusing the notations x_{ij}, the problem of finding $x_i \in V$ with $S_{m+1}(x_1, \ldots, x_m, X) = 0$ can now be reduced to solving either

$$\begin{cases} S_{m+1}(x_{11}, \ldots, x_{m1}, X) = 0 & \\ x_{ij} = x_{i,j-1}^q & i = 1, \ldots, m; j = 2, \ldots, n' \\ \sum_{j=0}^{n'} c_i x_{ij} = 0 & i = 1, \ldots, m \end{cases} \tag{2}$$

or

$$\begin{cases} S_{m+1}(x_{11}, \ldots, x_{m1}, X) = 0 & \\ x_{ij} = x_{i,j-1}^q - \alpha_{j-1} x_{i,j-1} & i = 1, \ldots, m; j = 2, \ldots, n' \\ 0 = x_{i,n'}^q - \alpha_{n'} x_{i,n'} & i = 1, \ldots, m. \end{cases} \tag{3}$$

The two systems have been suggested in [17,16,12]. Compared to polynomial systems arising from a Weil descent, both systems have the disadvantage to be defined over the field $\mathbb{F}_{q^n}$ but on the other hand they are much sparser and a priori easier to study. In fact, these systems are equivalent to polynomial systems arising from a Weil descent under linear changes of equations and variables, and in the univariate case ($m = 1$) their study has allowed to derive bounds on the corresponding Weil descent systems [16,12].

While Systems (2) and (3) can be solved with generic Gröbner basis algorithms, their simple structures might lead to better algorithms in the future. Most importantly for this article, they open the way to a generalization of previous algorithms to elliptic curves over prime fields.

3 Algebraic attacks on prime curves

3.1 Main idea

We replace the map L in Equation (1) by another algebraic or rational map over $\mathbb{F}_p$ which for a given m similarly satisfies the following two conditions

1. $|\{x \in \mathbb{F}_p | L(x) = 0\}| \approx \left|\left\{x \in \overline{\mathbb{F}_p} | L(x) = 0\right\}\right| \approx p^{1/m}$,
2. L can be written as a composition of low degree maps L_j.

Algorithm 1 Index calculus algorithm for prime curves

Require: p, E, P, Q defining a discrete logarithm problem
Ensure: discrete logarithm k such that $Q = kP$

1: Fix m
2: Find a suitable map L and its decomposition $L = \circ_{j=1}^{n'} L_j$
3: Define a factor base $\mathcal{F} = \{(x, y) \in E(K) | L(x) = 0\}$
4: Compute $\deg L + \Delta$ relations as follows

a. Pick $a, b \in \mathbb{F}_p$ randomly and compute $(X, Y) = aP + bQ$.
b. Construct and solve the system

$$\begin{cases} S_{m+1}(x_{11}, \ldots, x_{m1}, X) = 0 & \\ x_{i,j+1} = L_j(x_{i,j}) & i = 1, \ldots, m; j = 1, \ldots, n' - 1 \\ 0 = L_{n'}(x_{i,n'}) & i = 1, \ldots, m. \end{cases} \tag{4}$$

(If the L_j are rational maps, their denominators are put on the left-hand sides to obtain a polynomial system.)
c. For any solution found (modulo symmetries, namely permutations of the x_{i1}), if there are $P_i = (x_i, y_i) \in \mathcal{F}$ such that $\sum_i P_i = 0$, then store this relation.

5: Use linear algebra to solve the discrete logarithm

The resulting index calculus algorithm is summarized as Algorithm 1. For an optimal efficiency, the parameter m will have to be fixed depending on the cost of the relation search. At the moment, we have not investigated the existence of any algorithm better than Gröbner basis algorithms to solve System (4). The parameter Δ can be a priori fixed to 10; its aim is to account for linear depencies that may occur with a low probability between the relations.

The above conditions on L are such that: 1) most solutions of the system are defined over $\mathbb{F}_p$; 2) heuristically, we expect that the system has a constant probability to have a solution; 3) all the equations in the system have low degree. Note that System (4) is very similar to System (2) and System (3). We now show how these conditions can be satisfied, first for primes p such that $p - 1$ has a large smooth factor, and then for arbitrary primes.

3.2 Partial analysis

We consider a computation model where both arithmetic operations in $\mathbb{F}_p$ and elliptic curve scalar multiplications have unitary cost. This is of course a very rough approximation as scalar multiplications require a polynomial number of field operations, but the approximation will be sufficient for our purposes.

Let $T(E, m, L)$ be the time needed to solve System (4) for X chosen as in the algorithm. Let $P(p, m)$ be the *precomputation* time required to perform Step 2. The expected number of different solutions of the system in Steps 4b and 4c is about

$$\frac{(\deg L)^m}{m! \cdot p}.$$

Indeed, $|\mathcal{F}|$ is about $\deg(L)$ and $|E(K)|$ is about p. A given point (X, Y) is in the image of $\mathcal{F}^n \to E(K)$, $(P_r)_{r=1}^n \to \sum_{r=1}^n P_r$ about $\frac{(\deg L)^m}{m! \cdot p}$ times on average. The cost of Step 5 is

$$(\deg L)^\omega$$

where $2 < \omega \leq 3$ depends on the algorithm used for linear algebra. The total cost of the attack is therefore

$$P(p, m) + \frac{m! \cdot p}{(\deg L)^{m-1}} T(E, m, L) + (\deg L)^\omega.$$

Our algorithm will outperform generic discrete logarithm algorithms when this complexity is smaller than $p^{1/2}$. When $(\deg L)^m \approx m! \cdot p$, this will happen when one can solve $T(E, m, L)$ more efficiently than $p^{1/2-1/m}$.

3.3 Attack when $p - 1$ has a large smooth factor

Let us first assume that $p - 1 = r \prod_{i=1}^{n'} p_i$ where the p_i are not necessarily distinct primes, all smaller than B, and $\prod_{i=1}^{n'} p_i \approx p^{1/m}$. We do not impose any particular condition on r. We define V as the subgroup G of order $\prod_{i=1}^{n'} p_i$ in $\mathbb{F}_p^*$. We then set $L_j(x) = x^{p_j}$ for $j = 1, \ldots, n' - 1$, and $L_{n'}(x) = 1 - x^{p_{n'}}$. The function $L := \circ_{j=1}^{n'} L_j$ satisfies all the properties required.

Alternatively, we could also choose V as a coset aG of G, and adapt the maps accordingly.

Due to Pohlig-Hellman's attack [19], finite fields with smooth order, or an order with some large smooth factor, have long been discarded for the discrete logarithm problem over finite fields, but to the best of our knowledge there has been no similar result nor even warning with respect to elliptic curves. In fact, NIST curves use pseudo-Mersenne numbers and are therefore potentially more vulnerable to our approach than other curves. In particular, the prime number used to define NIST P-224 curve is such that

$$p - 1 = 2^{96} \cdot 3 \cdot 5 \cdot 17 \cdot 257 \cdot 641 \cdot 65537 \cdot 274177 \cdot 6700417 \cdot 67280421310721$$

hence it satisfies the prerequisites of our attack already for $m \geq 3$ and $B = 2$.

3.4 Generalization to arbitrary p

Let now p be an arbitrary prime number, in particular not necessarily of the previous form. Our second attack assumes the knowledge of an auxiliary elliptic curve $E'/\mathbb{F}_p$ with an order $N = r\prod_{i=1}^{n'} p_i$ where the p_i are not necessarily distinct primes, all smaller than B, and $\prod_{i=1}^{n'} p_i \approx p^{1/m}$. Note that the auxiliary curve is a priori unrelated to the curve specified by the elliptic curve discrete logarithm problem, except that it is defined over the same field. Let H be a subgroup of $E'(\mathbb{F}_p)$ of cardinality $\prod_{i=1}^{n'} p_i$. The set V will consist of the x-coordinates of all points $(x, y) \in E'(\mathbb{F}_p)$ in a coset of H. Let $\varphi : E' \to E'$ be the isogeny with kernel H. This isogeny can be efficiently written as a composition

$$\varphi = \varphi_{n'} \circ \ldots \circ \varphi_1$$

where $\deg \varphi_i = p_i$ and moreover all these isogenies can be efficiently computed using Vélu's formulae [24]. There exist polynomials ξ_j, ω_j, ψ_j such that

$$\varphi_j = \left(\frac{\xi_j(x)}{\psi_j^2(x)}, y\frac{\omega_j(x)}{\psi_j^3(x)} \right).$$

We then choose $L_j = \frac{\xi_j(x)}{\psi_j^2(x)}$ for $j = 1, \ldots, n'-1$ and $L_{n'} = \frac{\xi_{n'}(x)}{\psi_{n'}^2(x)} - \chi$, where χ is the x-coordinate of a point in the image of φ which is not 2-torsion. It is easy to check that the map $L = \circ_{j=1}^{n'} L_j$ then satisfies all properties required:

Lemma 1 *In the above construction,* $\{x \in \mathbb{F}_p | L(x) = 0\}$ *has size* $\prod_{i=1}^{n'} p_i$.

Proof: By construction, the isogeny φ has a kernel of size N/s, and so does any kernel coset. We claim that all the points in a coset have distinct x-coordinate if χ is not the x-coordinate of a point of order 2. Indeed, let $P_1 \neq P_2$ with $\varphi(P_1) = \varphi(P_2)$. If P_1 and P_2 have the same x-coordinate, then we have $P_2 = -P_1$ hence $\varphi(P_2) = \varphi(-P_1) = -\varphi(P_1)$. Therefore $\varphi(P_1) = -\varphi(P_1)$ has order 2. □

In Section 3.5 we discuss how the auxiliary curve E' can be found, first assuming that p has been fixed and cannot be changed, second assuming that we have some flexibility in choosing p as well.

3.5 Finding an auxiliary curve

We now consider the cost of Step 2 of our algorithm for general prime numbers. We propose two algorithms to perform this task: the first one just picks curves at random until one that has the good properties is found, the second one uses the theory of complex multiplication. As many applications will be using standardized curves such as NIST curves, these costs can be considered as precomputation costs in many applications. Finally, we show that they can be greatly reduced for an attacker who can choose the prime p.

Random curve selection The simplest method to perform the precomputation is to pick curves over $\mathbb{F}_p$ at random until one is found with a smooth enough order. To simplify the analysis, let us first consider a smoothness bound $B = 2$. The probability that the order of a random curve over $\mathbb{F}_p$ can be written as $N = 2^s \cdot r$ with $2^s \approx p^{1/m}$ is approximately $1/2^s \approx p^{-1/m}$, hence we expect to try about $p^{1/m}$ curves before finding a good one. Note that $p^{1/m}$ is essentially the size of the factor basis, hence the precomputation costs will always be dominated by at least the linear algebra costs in the whole index calculus algorithm. In practice we might be able to choose B bigger than 2, and this will make the precomputation cost even smaller, as shown by Table 1.

Table 1. Expected number of trials before finding a good curve, such that a factor at least $p^{1/m}$ is B-smooth. A number k in the table means that 2^k trials are needed on average. The numbers provided are for $|p| = 160$ and $|p| = 256$.

B/m	2	3	4	5
2	80.0	54.0	40.0	32.0
3	75.3	49.2	36.3	28.6
5	71.5	45.9	33.3	25.9
7	68.3	43.3	31.0	23.9
11	65.7	41.2	29.3	22.3
13	63.4	39.4	27.8	21.1
17	61.5	37.8	26.5	20.0

B/m	2	3	4	5
2	128.0	86.0	64.0	52.0
3	122.7	80.6	59.6	47.2
5	118.2	76.6	56.1	43.9
7	114.4	73.4	53.2	41.3
11	111.2	70.7	50.9	39.2
13	108.3	68.4	48.9	37.5
17	105.8	66.3	47.2	36.0

Complex multiplication The existence of a curve with N points over $\mathbb{F}_p$ within the Hasse-Weil bound is equivalent to the existence of an integer solution to the equation

$$(N + 1 - p)^2 - Df^2 = 4N$$

with $D < 0$ (see [4, Equation 4.3]). Once this solution is known, the curve can be constructed using the complex multiplication algorithm [3, p.30], provided however that the *reduced discriminant* D is not too large to compute the Hilbert class polynomial $H_D \bmod p$. To the best of our knowledge, the best algorithm for this task is due to Sutherland [23] and runs in quasi-linear time in $|D|$. Sutherland reports computations up to $|D| \approx 10^{13}$.

We can rewrite the above equation as $(p + 1 - N)^2 - Df^2 = 4p$ and try to solve it for some small D using Cornacchia's algorithm [5]. More precisely, we can solve the equation $x^2 - Dy^2 = 4p$ and check if the solution produces a number N which is divisible by a large enough smooth factor. This approach is relatively slow since the number of such N is relatively small. With $B = 2$, one needs to try about $p^{1/m}$ different values of D.

Faster precomputation for chosen p We now consider a different attack scenario, where p is not fixed but can be chosen by the attacker. In this setting,

we first construct a number N in such a way that we know its factorization, and that N has a large enough smooth factor. We then solve $x^2 - Dy^2 = 4N$ for some small $|D|$ (using the factorization of N). We check if the appropriate value for p is indeed prime, and if not we try a different small $|D|$. The probability of p being prime is about $1/\log(p) \approx 1/\log(N)$. This method allows to use much smaller $|D|$ and it will outperform previous methods in general.

We remark that this approach can potentially be applied to produce a sort of "back door" when choosing primes for elliptic curve cryptography standards. However, this seems unlikely for the following two reasons. First, as soon as a user is aware of the potential existence of such a back door, it can easily detect it by solving the above equation for the given p and all small values of D. Second, other equally useful auxiliary curves can be constructed in a time dominated by other steps of the index calculus algorithm.

4 Gröbner basis experiments

In this section we describe preliminary computer experiments to evaluate the complexity of relation search in our approach, and compare it to the binary case.

4.1 Experimental set-up

In the binary case, we selected random curves over $\mathbb{F}_{2^{2n_1}}$ and a fixed vector space $V = \langle 1, x, x^2, \ldots, x^{n_1} \rangle$, for $1 \leq n_1 \leq 11$.

For the attack of Section 3.3 we chose the smallest prime p such that 2^{n_1} divides $p-1$ with $p \geq 2^{2n_1}$, and V equal to the subgroup of order 2^{n_1} in $\mathbb{F}_p^*$.

For the attack of Section 3.4 we fixed $D = 7$. We selected parameters N and p such that there exists a curve of order N over $\mathbb{F}_p$, 2^{n_1} divides N, $N \in [2^{2n_1} - 2^{2n_1-2}; 2^{2n_1} + 2^{2n_1-1}]$ and N is the closest to 2^{2n_1} among those parameters. Using complex multiplication, we generated an elliptic curve E' over $\mathbb{F}_p$ with N rational points, and we computed a reduced Weierstrass model for this curve. We finally chose V as the projection on the x-coordinate of a coset of a subgroup of order 2^{n_1} of E', such that V had cardinality 2^{n_1}.

In all cases we selected a random (reduced Weierstrass model) curve over the field of consideration and a random point P on the curve. We then attempted to write $P = P_1 + P_2$ with P_i in the factor basis by reduction to polynomial systems and resolution of these systems with the Gröbner Basis routine of Magma. In the binary case we experimented on systems of the forms (2) and (3). In the other two cases we generated the systems as described in Section 3. We repeated all sets of experiments 100 times.

All experiments were performed on a CPU with 16-cores Intel Xeon Processor 5550, running at 2.67GHz with a L3 cache of 18MB. The Operating System was Linux Ubuntu 12.04.5 LTS with kernel version GNU/Linux 3.5.0-17-generic x86_64 and 24GB memory. The programming platform was Magma V2.18-5 in its 64-bit version.

4.2 Experimental results

In the tables below *nbsols* is the average number of solutions of the system, *Av. time* is the average time in seconds and *Max. mem* is the maximum amount of memory used. The values D_{av} and D_{av}^{corr} are the average values of two measures of the degree of regularity from Magma's verbose output. For D_{av} we take the largest "step degree" occurring during a Gröbner Basis computation. This corresponds to the degrees reported in [18]. For D_{av}^{corr} we correct that by removing any step in which no pair was reduced, as these steps should arguably not significantly impact the overall complexity of the algorithm. This corresponds to the degrees reported in [21] and [12].

Table 2. Binary case, SRA system

n_1	n	D_{av}	D_{av}^{corr}	nbsols	Av. time (s)	Max. mem (MB)
1	2	2.76	2.76	0.59	0.00	14
2	4	3.93	3.93	0.73	0.00	14
3	6	3.99	3.99	0.72	0.00	14
4	8	3.99	3.98	1.03	0.00	15
5	10	4.36	4.00	1.19	0.02	41
6	12	4.50	4.00	1.30	0.09	80
7	14	4.64	4.00	1.04	0.43	213
8	16	4.62	4.00	1.03	2.21	622
9	18	4.56	4.00	0.78	9.27	1555
10	20	5.14	4.00	1.26	38.83	4170
11	22	4.93	4.00	0.94	207.72	53173

Table 3. Binary case, System (2)

n_1	n	D_{av}	D_{av}^{corr}	nbsols	Av. time (s)	Max. mem (MB)
1	2	2.60	2.60	0.95	0.00	14
2	4	3.93	3.93	0.54	0.00	14
3	6	4.00	4.00	0.96	0.00	14
4	8	4.38	4.00	1.14	0.00	15
5	10	4.35	3.99	1.06	0.01	25
6	12	4.39	4.00	0.98	0.05	18
7	14	4.32	4.00	0.91	0.20	19
8	16	4.66	4.00	1.18	2.04	24
9	18	4.74	4.00	1.18	4.90	34
10	20	4.62	4.00	0.98	39.00	65
11	22	4.70	4.00	1.00	4989.96	256

Based on this (limited) set of experiments we make the following observations:

Table 4. Prime case, $p-1$ subgroups

n_1	p	D_{av}	D_{av}^{corr}	nbsols	Av. time (s)	Max. mem (MB)
1	5	4.00	4.00	0.59	0.00	9
2	17	4.00	4.00	0.79	0.00	9
3	73	4.00	4.00	0.84	0.00	9
4	257	4.01	4.00	1.14	0.00	9
5	1153	4.48	4.00	1.34	0.02	10
6	4289	5.00	5.00	1.08	0.13	13
7	17921	5.36	5.00	0.99	1.14	17
8	65537	5.36	5.00	0.96	9.09	35
9	262657	5.78	5.00	1.06	59.87	98
10	1051649	6.36	6.00	0.96	454.79	501
11	4206593	6.29	6.00	0.76	4975.07	2266

Table 5. Prime case, isogeny kernel

N	p	D_{av}	D_{av}^{corr}	nbsols	Av. time (s)	Max. mem (MB)
$2^1 \cdot 4$	11	4.00	4.00	0.25	0.00	11
$2^2 \cdot 7$	29	5.00	5.00	0.44	0.00	11
$2^3 \cdot 7$	71	5.00	5.00	0.91	0.01	11
$2^4 \cdot 22$	359	5.00	5.00	0.52	0.01	11
$2^5 \cdot 29$	967	5.00	5.00	0.97	0.03	11
$2^6 \cdot 53$	3467	5.42	5.00	1.23	0.15	12
$2^7 \cdot 106$	13619	5.47	5.00	1.17	1.16	18
$2^8 \cdot 203$	52291	5.42	5.00	1.13	9.08	34
$2^9 \cdot 414$	212587	5.92	5.00	1.19	51.87	95
$2^{10} \cdot 791$	811763	6.44	6.00	1.04	438.57	367
$2^{11} \cdot 1548$	3173683	6.45	6.00	1.06	5163.46	1945

1. The corrected version of the degree of regularity is a very stable measure: except for very small parameters, no variation was observed within any set of 100 experiments.
2. In our experiments, systems in the form (2) require much less memory than the corresponding SRA systems.
3. Timing comparison is less clear: while systems in the form (2) are more efficient up to $n_1 = 10$, SRA systems are much better at $n_2 = 11$.
4. The degrees of regularity, time and memory requirements are similar in the subgroup and isogeny versions of our attack.
5. The degrees of regularity, time and memory requirements seem to increase a bit faster in the prime case than in the binary case in general.

According to Bardet [1, Prop 4.1.2], homogeneous semi-generic systems with n equations of degree 2 and 1 equation of degree 4 in n variables have a degree of regularity equal to $(3+n)/2$. In all our experiments we observed a much smaller dependency in n of the degree of regularity, suggesting that the systems occurring

in our attacks are easier to solve than semi-generic systems with comparable parameters.

5 Conclusion, further work and perspectives

Our algorithms generalize previous index calculus attacks from binary curves to prime curves, and therefore considerably increase their potential impact. All attacks including ours (implicitly) reduce the relation search in index calculus algorithms to an instance of the following problem:

Problem 1 (Generalized root-finding problem) *Given a finite field K, given $f \in K[X_1, \ldots, X_m]$, and given $L \in K(X)$, find $X_i \in K$ such that $f(X_1, \ldots, X_m) = 0$ and $L(X_i) = 0$ for all i.*

We have suggested to focus on special polynomials L, which can be written as compositions of low degree maps, so that the generalized root-finding problem can be reduced to a polynomial system similar to "SRA systems" [17, Section 6], and then solved using Gröbner basis algorithms. Our computer experiments suggest that the resulting systems are a bit harder to solve than the corresponding systems in binary cases, but easier to solve than generic systems of comparable parameters.

The attacks are not practical at the moment and we do not know their asymptotic complexity. Still, we believe that they do unveil potential vulnerabilities that cryptanalysts need to study further. In particular, we showed that the standardized curve NIST P-224 satisfies the requirements of our first attack.

Following a suggestion by the PKC 2016 committee, we have also compared our approach with a variant of Semaev's original attack using Groebner basis algorithms to solve the system $S(x_1, x_2, X) = 0$, $L(x_1) = 0$, $L(x_2) = 0$ with $L(x) = \prod_{\alpha<B}(x - \alpha)$. Intriguingly, our preliminary results show that these systems are easier to solve than ours using Groebner basis algorithms on similar parameters. This can perhaps be explained by the much smaller number of variables, and may either suggest that our approach is unlikely to be efficient asymptotically, or that Semaev's original attack should be revisited from an algebraic perspective.

Important open problems include providing a satisfactory theoretical explanation for our experiments, and predicting the complexity of all algorithms for large parameters.

An even more important problem is to design a dedicated algorithm for the generalized root-finding problem, which would not rely on Gröbner basis algorithms at all. It is worth noticing that the Weil descent and Gröbner basis approach, when applied to classical root-finding problems (where f is univariate and $L(x) = x^{|K|} - x$), provides an algorithm with complexity exponential in $O(\log n \cdot \deg f)$ under a somewhat controversial heuristic assumption, whereas the best algorithms for this problem have a provable complexity exponential in $O(\log n + \deg f)$. A similar improvement for the above generalized version of the root-finding problem will greatly impact elliptic curve cryptography.

Acknowledgement We thank Steven Galbraith, Sze Ling Yeo and Andreas Enge for discussions and suggestions related to this work. We also thank the program committee for their numerous and helpful suggestions. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 307937, the Engineering and Physical Sciences Research Council grant EP/J009520/1, and GCHQ through a research grant. The second author was initially funded by TL@NTU, but is currently affiliated to UCI.

References

1. M. Bardet. *Etude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* PhD thesis, Université Paris 6, 2004.
2. E. R. Berlekamp. *Algebraic coding theory.* Aegean Park Press, Laguna Hills, CA, USA, 1984.
3. R. Bröker. Constructing supersingular elliptic curves.
4. R. Bröker. *CONSTRUCTING ELLIPTIC CURVES OF PRESCRIBED ORDER.* PhD thesis, University of Leiden, 2006.
5. G. Cornacchia. Su di un metodo per la risoluzione in numeri interi dell' equazione $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$. *Giornale di Matematiche di Battaglini*, 46:33–90, 1903.
6. B. den Boer. Diffie-Hellman is as strong as discrete log for certain primes. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, volume 403 of *Lecture Notes in Computer Science*, pages 530–539. Springer, 1988.
7. C. Diem. On the discrete logarithm problem in elliptic curves. *Compositio Mathematica*, 147:75–104, 2011.
8. C. Diem. On the discrete logarithm problem in elliptic curves II. *Algebra & Number Theory*, 7:1281–1323, 2013.
9. J.-C. Faugère, L. Perret, C. Petit, and G. Renault. Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 27–44. Springer, 2012.
10. S. D. Galbraith and S. W. Gebregiyorgis. Summation polynomial algorithms for elliptic curves in characteristic two. In W. Meier and D. Mukhopadhyay, editors, *Progress in Cryptology - INDOCRYPT 2014 - 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings*, volume 8885 of *Lecture Notes in Computer Science*, pages 409–427. Springer, 2014.
11. P. Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *J. Symb. Comput.*, 44(12):1690–1702, 2009.
12. M. A. Huang, M. Kosters, and S. L. Yeo. Last fall degree, HFE, and Weil descent attacks on ECDLP. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 581–600, 2015.
13. Y.-J. Huang, C. Petit, N. Shinohara, and T. Takagi. Improvement of Faugère et al.'s method to solve ECDLP. In K. Sakiyama and M. Terada, editors, *IWSEC*, volume 8231 of *Lecture Notes in Computer Science*, pages 115–132. Springer, 2013.
14. H. W. Lenstra. Factoring integers with elliptic curves. *Ann. of Math.*, 126:649–673, 1987.

15. U. M. Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete algorithms. In Y. Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pages 271–281. Springer, 1994.
16. C. Petit. Bounding HFE with SRA. `http://perso.uclouvain.be/christophe.petit/files/SRA_GB.pdf`, 2014.
17. C. Petit. Finding roots in $GF(p^n)$ with the successive resultant algorithm. *LMS Journal of Computation and Mathematics*, 17A:203–217, 2014.
18. C. Petit and J.-J. Quisquater. On polynomial systems arising from a Weil descent. In X. Wang and K. Sako, editors, *Asiacrypt*, volume 7658 of *Lecture Notes in Computer Science*, pages 451–466. Springer, 2012.
19. S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance (corresp.). *IEEE Transactions on Information Theory*, 24(1):106–110, 1978.
20. I. Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. Cryptology ePrint Archive, Report 2004/031, 2004. `http://eprint.iacr.org/`.
21. I. Semaev. New algorithm for the discrete logarithm problem on elliptic curves. Cryptology ePrint Archive, Report 2015/310, 2015. `http://eprint.iacr.org/`.
22. M. Shantz and E. Teske. Solving the elliptic curve discrete logarithm problem using Semaev polynomials, Weil descent and Gröbner Basis methods - an experimental study. In *Number Theory and Cryptography - Papers in Honor of Johannes Buchmann on the Occasion of His 60th Birthday*, pages 94–107, 2013.
23. A. V. Sutherland. Computing Hilbert class polynomials with the Chinese remainder theorem. *Math. Comput.*, 80(273):501–538, 2011.
24. J. Vélu. Isogénies entre courbes elliptiques. *Communications de l'Académie royale des Sciences de Paris*, 273:238–241, 1971.