Improving the Complexity of Index Calculus Algorithms in Elliptic Curves over Binary Fields

Jean-Charles Faugère[1]**, Ludovic Perret[1]**, Christophe Petit[2]*, and Guénaël Renault[1]**

[1] UPMC, Université Paris 06, LIP6
INRIA, Centre Paris-Rocquencourt, PolSys Project-team
CNRS, UMR 7606, LIP6
4 place Jussieu, 75252 Paris, Cedex 5, France
`Jean-Charles.Faugere,Ludovic.Perret,Guenael.Renault@lip6.fr`

[2] UCL Crypto Group,
Université catholique de Louvain
Place du levant 3, 1348 Louvain-la-Neuve, Belgium
`christophe.petit@uclouvain.be`

Abstract. The goal of this paper is to further study the index calculus method that was first introduced by Semaev for solving the ECDLP and later developed by Gaudry and Diem. In particular, we focus on the step which consists in decomposing points of the curve with respect to an appropriately chosen factor basis. This part can be nicely reformulated as a purely algebraic problem consisting in finding solutions to a multivariate polynomial $\mathbf{f}(\mathbf{x_1}, \ldots, \mathbf{x_m}) = \mathbf{0}$ such that $\mathbf{x_1}, \ldots, \mathbf{x_m}$ all belong to some vector subspace of $\mathbb{F}_{2^n}/\mathbb{F}_2$. Our main contribution is the identification of particular structures inherent to such polynomial systems and a dedicated method for tackling this problem. We solve it by means of Gröbner basis techniques and analyze its complexity using the multi-homogeneous structure of the equations. A direct consequence of our results is an index calculus algorithm solving ECDLP over any binary field $\mathbb{F}_{2^n}$ in time $O(2^{\omega\, t})$, with $t \approx n/2$ (provided that a certain heuristic assumption holds). This has to be compared with Diem's [14] index calculus based approach for solving ECDLP over $\mathbb{F}_{q^n}$ which has complexity $\exp\big(O(n \log(n)^{1/2})\big)$ for $q = 2$ and n a prime (but this holds without any heuristic assumption). We emphasize that the complexity obtained here is very conservative in comparison to experimental results. We hope the new ideas provided here may lead to efficient index calculus based methods for solving ECDLP in theory and practice.

Keywords: Elliptic Curve Cryptography, Index Calculus, Polynomial System Solving.

* F.R.S.-FNRS postdoctoral research fellow at Université Catholique de Louvain, Louvain-la-Neuve. This work was partly supported by the Belgian State's IAP program P6/26 BCRYPT.

** This work was partly supported by the Commission of the European Communities through the ICT program under contract ICT-2007-216676 (ECRYPT-II) and by the French ANR under the Computer Algebra and Cryptography (CAC) project (ANR-09-JCJCJ-0064-01).

1 Introduction

Elliptic curves were independently introduced to cryptography by Miller and Koblitz in 1985 [37,32]. The security of curve-based cryptosystems often relies on the difficulty of solving the well-known Discrete Logarithm Problem on Elliptic Curves (ECDLP). Given a finite cyclic group $G = \langle P \rangle$ and given an element Q in G, the discrete logarithm problem asks for an integer k such that $Q = kP$. For an elliptic curve E defined over a finite field K, the group G can chosen to be the set $E(K)$ of rational points on E.

One of the main method for solving (EC)DLP is *Index Calculus.* This approach, which was first introduced by Kraitchik [33] and later optimized by Adleman [1], can be seen as a general framework [15]. To summarize, Index Calculus algorithms are composed of the following three steps:

1. ***Factor Basis*** **definition.** Identify a subset $\mathcal{F} = \{\pi_1, \dots, \pi_s\} \subset G$.
2. ***Sieving*** **step.** Collect more than $\#\mathcal{F}$ relations of the form $aP + bQ = \sum_{i=1}^{s} e_i \pi_i$ where a, b are random integers.
3. ***Linear Algebra*** **step.** From these relations, construct a matrix with a non-trivial kernel. Then, find a non-trivial vector in this kernel and deduce k (a discrete logarithm) from such vector.

While the last step is independent of the choice of G, the efficiency of the first two steps relies on specific properties of the group. Depending on the ability to find a factor basis together with an algorithm for computing relations during the sieving step, the index calculus method may achieve a *subexponential* complexity. For instance, subexponential algorithms have been obtained for multiplicative groups of finite fields [3,2,4,30] and for Jacobian groups of hyperelleptic curves with large genus [3,27,26].

Related Works and Contributions. Our results are related to the sieving step of the ECDLP index calculus method proposed by Semaev [40] and later developed by Gaudry [28] and Diem [13,14]. The main idea introduced by Semaev is the use of so-called *summation polynomials.* As soon as a factor basis is fixed, summation polynomials can be used for sieving elements of an elliptic curve E and thus an index calculus method follows. The main problem to get an efficient index calculus is to find a factor basis together with an efficient algorithm for solving summation polynomials. During the ECC conference following publication of Semaev's result, Gaudry and Diem independently proposed such solutions. More precisely, when E is defined over an extension $\mathbb{F}_{q^n}$ with $n > 1$ a composite integer, Gaudry [28] proposed to use the following set $\mathcal{F} = \{(\mathbf{x}, \mathbf{y}) \in E(\mathbb{F}_{q^n}) \mid \mathbf{x} \in \mathbb{F}_q\}$ as a factor basis and provides an algorithm for solving the ECDLP with a complexity better than generic methods. Next, the problem of finding $P_1, \dots, P_m$ in $\mathcal{F}$ such that $R = P_1 + \cdots + P_m$ for a given $R \in E$ is reduced to solve the equation $\mathbf{S_{m+1}}((P_1)_x, \dots, (P_m)_x, R_x) = 0$, where $\mathbf{S_r}$ is the r-th Semaev's summation polynomial [40] and $(P)_x$ stands for the x-coordinate of P. Diem proposed a generalization of this approach by considering (in a simpler form here) the factor basis $\mathcal{F}_V := \{(\mathbf{x}, \mathbf{y}) \in E(K) \mid \mathbf{x} \in V\}$, with V a vector subspace of $\mathbb{F}_{q^n}/\mathbb{F}_q$. Thus, the main computational tool for sieving here is an algorithm solving efficiently a specific polynomial system.

Recently, Diem presented in [14] new complexity results for this generalization. He succeeds to prove – without any heuristic assumption – some subexponential complexity results for solving ECDLP. But, for $q = 2$ and n is a prime – the setting considered here – he has an index calculus algorithm of exponential complexity $e^{O\left(n \log(n)^{1/2}\right)}$. The polynomial systems occurring in [14] are solved with a geometrical algorithm proposed by Rojas [39]. Whilst such algorithm has a good complexity estimate, it is well known that its hard to implement it in practice.

In this work, we focus on the specific case $q = 2$ and n prime. We show that the polynomial systems occurring have a very specific structure. We provide a new (heuristic) algorithm taking advantage of the structure for the sieving step. We focus our study on the following point decomposition problem related to some vector space:

Problem 1 (Point Decomposition Problem associated to a vector space V). Let V be a vector space of $\mathbb{F}_{2^n}/\mathbb{F}_2$. Given a point $R \in E(\mathbb{F}_{2^n})$, the problem is to find – if any – m points $P_1, \ldots, P_m$ such that $R = P_1 + \cdots + P_m$ with the additional constraint that $(P_i)_x \in V$ for all $i \in \{1, \ldots, m\}$.

This problem can be reduced to a polynomial system solving problem using Semaev's summation polynomials (or using any other polynomial system modeling). Here, we show that the *multi-homogeneous* structure of the system constructed from Semaev's summation polynomials can be used to design a Gröbner based algorithm. To be more precise, we design an efficient algorithm for:

Problem 2. Let $t \geq 1$, V be a vector space of $\mathbb{F}_{2^n}/\mathbb{F}_2$ and $\mathbf{f} \in \mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_m}]$ be any multivariate polynomial of degree bounded by $2^t - 1$ in each variable. The problem is to find $(\mathbf{z_1}, \ldots, \mathbf{z_m}) \in V^m$ such that $\mathbf{f}(\mathbf{z_1}, \ldots, \mathbf{z_m}) = \mathbf{0}$.

Since $\mathbb{F}_{2^n}$ is a vector space over $\mathbb{F}_2$, $\mathbf{f}$ can be rewritten (or deployed) as a polynomial system of m equations over $\mathbb{F}_2$ and then can be solved using Gröbner bases algorithms. The prominent observation is to remark that this system is (affine) *multi-homogeneous.* While the complexity of solving bi-linear systems using Gröbner bases – that is to say polynomials of bi-degree $(1, 1)$ – is now well understood [24], the general case is not known. Consequently, we propose a simple *ad-hoc* algorithm to take advantage of the multihomogeneous structure. This is of independent interest in the more general context of computer algebra.

The main idea is to show that starting from the unique equation $\mathbf{f} = \mathbf{0}$, we can generate many low-degree equations by deploying the equations $\mathbf{mf} = \mathbf{0}$ over $\mathbb{F}_2$ for a large number of appropriately chosen monomials $\mathbf{m}$. Another main difference with unstructured polynomials is that – in some degree d – the number of monomials occurring in the polynomials coming from $\mathbf{mf}$ is much smaller than the number of all possible monomials in degree d. Indeed, due to the choice of $\mathbf{m}$, the monomials occurring in the equations constructed from $\mathbf{mf}$ have still a multi-homogeneous structure and their degrees are well controlled.

As usual, to estimate the maximum degree D reached during the computation we study the number of equations minus the number of monomials. When this number is > 0, and under a reasonable linear independence assumption confirmed by our experimental results, the computation is finished. Using the structure of the polynomials, we prove that this degree D is much smaller than expected; assuming that a reasonable heuristic is true. It is worth noticing that although we describe our algorithm as a linearization method [35], Gröbner basis algorithms like F_4 or F_5 [16,17] can be advantageously used in practice to solve the corresponding polynomial systems. The algorithm presented in this paper, together with its complexity analysis, can thus be understood as a method to (heuristically) bound the complexity of computing the corresponding Gröbner basis similarly to the Macaulay's bound obtain by Lazard [34] to bound the complexity of Gröbner bases in the general case. More precisely, we obtain:

Theorem 2. *Assuming some linear independence assumption (see Assumption 1, p. 10), Problem 2 can asymptotically be solved in time $O(2^{\omega\tau})$ and memory $O(2^{2\tau})$, where ω is the linear algebra constant and $\tau \approx n/2$. Under the same hypothesis, there exists an index calculus based algorithm solving ECDLP over $\mathbb{F}_{2^n}$ in the same time complexity.*

The index calculus algorithm presented here has a better complexity than the one proposed recently by Diem [14]. Moreover, we propose a novel approach for solving the sieving step. We consider that it is a major open challenge to further exploit the intrinsic algebraic structure of Problem 2 using Gröbner bases algorithms. The complexity obtained here for solving ECDLP is still exponential. We hope that the structures identified here can be further used to get a complexity better than generic algorithms (see preliminary experiments in Section 5.2) or to get a subexponential algorithm in a near future. Finally, we emphasize that the complexity analysis of our algorithm relies on a *heuristic* assumption on the rank of a linearized system. The validity of this assumption was experimentally checked (see Section 5.1). Whilst we pushed the experiments as far as possible, we pointed out that – due to the size of the systems involved – it is very difficult to verify experimentally the assumption for large parameters. Consequently, it is an open issue to prove Assumption 1.

Outline. The remaining of this paper is organized as follows. In Section 2, we detail our notations and we provide some background on Gröbner bases. In Section 3, we introduce and analyze our algorithm for solving Problem 2. In Section 4, we apply our new result to the ECDLP over binary fields. In Section 5, we present experimental results to give first evidences for Assumption 1. Section 6 concludes the paper and introduces future extensions of our work.

2 Preliminaries

In this section, we introduce definitions, notations and recall well known results concerning polynomial system solving.

2.1 Definition and Notation

Let $\mathbb{F}_2$ be the finite field of cardinality 2. We will consider a degree n extension $\mathbb{F}_{2^n}$ of $\mathbb{F}_2$. We will often see $\mathbb{F}_{2^n}$ as an n dimensional vector space over $\mathbb{F}_2$. Let $\{\theta_1, \ldots, \theta_n\}$ be a basis of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$. We will use bold letters for elements, variables and polynomials over $\mathbb{F}_{2^n}$ and normal letters for elements, variables and polynomials over $\mathbb{F}_2$. If $x_1, \ldots, x_m$ are algebraic independent variables over a finite field $\mathbb{K}$, we write $R = \mathbb{K}[x_1, \ldots, x_m]$ for the polynomial ring in those variables. Given a set of polynomials $\{f_1, \ldots, f_\ell\} \in R$, the *ideal* generated by this set will be denoted by $\langle f_1, \ldots, f_\ell \rangle \subset R$. We write $\mathrm{Res}_{x_i}(f_1, f_2)$ for the *resultant* of $f_1 \in R$ and $f_2 \in R$ with respect to the variable x_i. A power product, i.e. $\prod_{i=1}^{k} x_i^{e_i}$ where $e_i \in \mathbb{N}$, is called a *monomial.* Finally, we introduce a structure which will be very useful in this paper.

Definition 1 ([24]). *Let $X_1 \cup X_2 \cup \cdots \cup X_t = \{x_1, \ldots, x_m\}$ be a partition of the variables set. We shall say that a polynomial $f \in \mathbb{K}[x_1, \ldots, x_m] = \mathbb{K}[X_1, \ldots, X_t]$ is* multi-homogeneous *of* multi-degree $(d_1, d_2, \ldots, d_t)$ *if:*

$$\forall (\alpha_1, \ldots, \alpha_t) \in \mathbb{K}^t,\ f(\alpha_1 X_1, \ldots, \alpha_t X_t) = \alpha_1^{d_1} \cdots \alpha^{d_t} f(X_1, \ldots, X_t).$$

For all $i, 1 \le i \le t$, let $X_i = \{x_{i,1}, \ldots, x_{i,n_i}\}$. We shall say that f is affine multi-homogeneous *if there exists $f^h \in \mathbb{K}[X_1, \ldots, X_t]$ a multi-homogeneous polynomial of same degree such that when one replaces (homogenization) variables x_{i,n_i} by 1 we obtains f, i.e.: $f(x_{1,1}, \ldots, x_{1,n_1-1}, \ldots, x_{t,1}, \ldots, x_{t,n_t-1})$ is equal to $f_i^h(x_{1,1}, \ldots, x_{1,n_1-1}, 1, \ldots, x_{t,1}, \ldots, x_{t,n_t-1}, 1)$. Finally, we shall say that f has a* multi-homogeneous structure *if it is multi-homogeneous or affine multi-homogeneous. A system of equation has a* multi-homogeneous structure *if each equation has a multi-homogeneous structure (the equations can have different multi-degrees).*

Given a number $e = \sum_{i=0}^{\infty} e_i 2^i$ with $e_i \in \{0, 1\}$, we define its *Hamming weight* as the number of non-zero elements in its binary expansion, i.e. $W(e) := \sum_{i=0}^{\infty} e_i$. We write $\binom{n}{k}$ for the number of choices of k elements among a set of n elements without repetition. We write O for the "big O" notation: given two functions f and g of n, we say that $f = O(g)$ if there exist $N, c \in \mathbb{Z}^+$ such that $n > N \Rightarrow f(n) \le cg(n)$. The notation log stands for the binary logarithm. Finally, we write ω for the *linear algebra constant.* Depending on the algorithm used for linear algebra, we have $2.376 \le \omega \le 3$.

2.2 Gröbner Bases [10]

Recent methods such as Faugère's F_4 and F_5 [16,17] algorithms reduce Gröbner basis computation to Gaussian eliminations on several matrices. The link between linear algebra and Gröbner bases has been established by Lazard [34]. He showed that computing a Gröbner basis is equivalent to perform Gaussian elimination on the so-called *Macaulay matrices* as defined below:

Definition 2 (Macaulay Matrix [35,36]).

$$\begin{array}{cc} & m_1 > m_2 > \dots \\ \vdots & \\ t_{i,j} f_i & \begin{pmatrix} & & \\ c_{i,j}^1 & c_{i,j}^2 & \dots \\ & & \end{pmatrix} \end{array}$$

Let $F = \{f_1, \dots, f_\ell\} \subset R$ be a set of polynomials of degree $\leq d$. Let $\mathcal{B} = \{m_1 > m_2 > \cdots\} \subset R$ be the sorted set (w.r.t. a fixed monomial ordering) of degree $\leq d$ monomials. The set $\mathcal{B}$ is a basis of the vector space of degree $\leq d$ polynomials in R. The Macaulay matrix $\mathcal{M}_d(F)$ *of degree d is defined as follows. We consider all the polynomials $t_{i,j} f_i$ of degree $\leq d$ with $t_{i,j} \in \mathcal{B}$ and $f_i \in F$. Rows of $\mathcal{M}_d(F)$ correspond to the coefficients vectors $(c_{i,j}^1, c_{i,j}^2, \dots)$ of these polynomials $t_{i,j} f_i = \sum_k c_{i,j}^k m_k$ with respect to the basis $\mathcal{B}$.*

Precisely, Lazard [34] proved the following fundamental result:

Theorem 1. *Let $F = \{f_1, \dots, f_\ell\} \subset R$. There exists a positive integer D for which Gaussian elimination on all matrices $\mathcal{M}_1(F), \mathcal{M}_2(F), \dots, \mathcal{M}_D(F)$ computes a Gröbner basis of $\langle f_1, \dots, f_\ell \rangle$.*

F_4 [16] can be seen as another way to use linear algebra without knowing an a priori bound. It successively constructs and reduces matrices until a Gröbner basis is found. The same is true for F_5 when considered in "F_4-style" [25].

It is clear that an important parameter in Gröbner basis computation is the maximal degree D reached during the computation. This maximal degree is called the *degree of regularity*. However, it is a difficult problem to estimate *a priori* the degree of regularity. This degree is known and well mastered for specific families of systems called regular and semi-regular [5,6,7]. It is classical to assume that the regularity of regular/semi-regular systems provides an extremely tight upper bound on the regularity of random system of equations (most of the times, we have equality). For example, the regularity degree of a regular sequence $f_1, \dots, f_\ell \in R$ (with $\ell \leq n$) is $D = 1 + \sum_{i=1}^{\ell} (\deg(f_i) - 1)$. Ideals with special structures – typically arising in cryptography – may have a much lower regularity degree, hence a much better time complexity. This has permitted Gröbner bases techniques to successfully attack many cryptosystems (e.g. the *Hidden Field Equation* cryptosystem (HFE) [38,31,18,29], Multi-HFE [9]). In many cases, the algebraic systems appearing in these applications were not generic and could be solved more efficiently than generic systems, sometimes using *dedicated* Gröbner basis algorithms.

In our case, the algebraic system considered in Problem 2 has a *multi-homogeneous* structure (more precisely *multi-linear*). Interestingly, the formal study of computing Gröbner basis of multi-homogeneous systems has been initiated by Faugère, Safey El Din and Spaenlehauer for bilinear systems [24] leading already to new cryptanalytic results [19,23,21]. However, the general case (more blocks, larger degrees) remains to be investigated.

As a consequence, we design in this paper a simple *ad hoc* algorithm to take advantage of the multi-homogeneous structure arising in Problem 2. Basically, the idea is to generate a submatrix of the Macaulay matrix at a degree D_{Lin} which allows to linearize the system derived from Problem 2. To do so, we strongly exploit the multi-homogeneous structure of the equations. The fundamental remark is that many low-degree relations exist and can be explicitly predicted. Moreover, the number of columns in the Macaulay matrix is less than usual. This is due to the fact that all monomials occurring have still the multi-homogeneous structure. Roughly, this allows to predict many zero columns for a suitably chosen subset of the rows. In section 3.2, we derive a bound on the degree D_{Lin} which needs to be considered.

It is worth noticing that although we describe our algorithm as a linearization method, computing a Gröbner basis with F_4 or F_5 [16,17] can be advantageously used in practice to solve the corresponding polynomial systems. The D_{Lin} obtained has to be understood as an upper bound on the real regularity degree of the system. Any new result on multi-homogeneous systems would allow to improve the complexity of solving Problem 2 (partial results are presented in Section 5.2).

3 Solving Multivariate Polynomials with Linear Constraints

In this section, we describe the main result of this paper: an algorithm for solving Problem 2. Let V be a $\mathbb{F}_2$-vector subspace $\subset \mathbb{F}_{2^n}$ of dimension n' and let $\mathbf{f} \in \mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_m}]$ be a multivariate polynomial with degree $< 2^t$ in each variable. We want to solve $\mathbf{f}(\mathbf{x_1}, \ldots, \mathbf{x_m}) = \mathbf{0}$ under the *linear constraints* $\mathbf{x_1}, \ldots, \mathbf{x_m} \in V$. To tackle this problem, we generalize the algorithm and analysis of [22] from the multi-linear case ($t = 1$) to arbitrary values of t. From now on, we assume that $m \cdot n' \approx n$ so that the problem has about one solution on average.

3.1 Modeling the Linear Constraints

Let $\{\nu_1, \ldots, \nu_{n'}\} \subset \mathbb{F}_{2^n}$ be a basis of V as a $\mathbb{F}_2$-vector space. Let $y_{i,j}$ be $m \cdot n'$ variables defined by $x_i = \nu_1 y_{i,1} + \nu_2 y_{i,2} + \cdots + \nu_{n'} y_{i,n'}$. We apply a *Weil descent* to Problem 2 (see e.g. [11, Chapter 7]). By replacing the variables x_i in the polynomial $\mathbf{f}$, we get a new polynomial $\mathbf{f}_V \in \mathbb{F}_{2^n}[y_{1,1}, \ldots, y_{m,n'}]$ with $m \cdot n'$ variables. The linear constraints on $\mathbf{f}$ are translated to *Galoisian constraints* by constraining the solutions of $\mathbf{f}_V$ to $\mathbb{F}_2$. Using the *field equations*, $\mathbf{f}_V$ is viewed more precisely as a a polynomial in the affine algebra $\mathcal{A}(\mathbb{F}_{2^n}) := \mathbb{F}_{2^n}[y_{1,1}, \ldots, y_{m,n'}]/\langle \mathcal{S}_{\mathrm{fe}} \rangle$, where $\langle \mathcal{S}_{\mathrm{fe}} \rangle$ is the 0-dimensional ideal generated by the field equations:

$$\mathcal{S}_{\mathrm{fe}} = \{y_{i,j}^2 - y_{i,j}\}_{1 \le i \le m}^{1 \le j \le n'}.$$

Problem 2 is then equivalent to compute a Gröbner basis of the ideal $\langle \mathbf{f}_V, \mathcal{S}_{\mathrm{fe}} \rangle \subset \mathbb{F}_{2^n}[y_{1,1}, \ldots, y_{m,n'}]$. It is generally more efficient to consider its resolution over $\mathbb{F}_2$. To do so, we consider $\mathcal{A}(\mathbb{F}_{2^n})$ as a module over $\mathcal{A}(\mathbb{F}_2) = \mathbb{F}_2[y_{1,1}, \ldots, y_{m,n'}]/\langle \mathcal{S}_{\mathrm{fe}} \rangle$ whose basis is $\{\theta_1, \ldots, \theta_n\}$. We consider $\mathbf{f}_V$ as an element of $\mathcal{A}(\mathbb{F}_{2^n})$ and we *deploy* it as a $\mathcal{A}(\mathbb{F}_2)$-linear combination of the basis $\{\theta_1, \ldots, \theta_n\}$. Namely:

$$\mathbf{f_V} = [\mathbf{f_V}]_1^{\downarrow} \theta_1 + [\mathbf{f_V}]_2^{\downarrow} \theta_2 + \cdots + [\mathbf{f_V}]_n^{\downarrow} \theta_n \qquad (1)$$

for some $[\mathbf{f_V}]_1^\downarrow, \ldots, [\mathbf{f_V}]_n^\downarrow \in \mathcal{A}(\mathbb{F}_2)$ that depend on $\mathbf{f}$ and the vector subspace V. Due to the linear independence of the $\theta_1, \ldots, \theta_n$, Problem 2 is equivalent to solve:

$$\mathcal{S}_{\text{alg}}: \quad [\mathbf{f_V}]_1^\downarrow = [\mathbf{f_V}]_2^\downarrow = \cdots = [\mathbf{f_V}]_n^\downarrow = 0. \tag{2}$$

In order to solve $\mathcal{S}_{\text{alg}}$, we will generate many new equations by deploying multiples of $\mathbf{f} \in \mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_m}]$ over the vector subspace V. The key point of this strategy is the existence of abnormally high number of low-degree equations arising as algebraic combinations of the equations in (2).

From now on, we represent the classes of polynomials $\mathbf{g_V} \in \mathcal{A}(\mathbb{F}_{2^n})$ and $[\mathbf{g_V}]_i^\downarrow \in \mathcal{A}(\mathbb{F}_2)$ corresponding to $\mathbf{g} \in \mathbb{F}_{2^n}[x_1, \ldots, x_m]$ by their minimal elements, in other words by their normal forms modulo $\mathcal{I}_{\text{fe}}$ (whose generators form a Gröbner basis). By abuse of notation, we use the same symbol for a class and its minimal representative in the underlying polynomial ring. When the context is not clear, we precise the algebra where the element is lying.

3.2 Low-Degree Equations

Let $e_1, \ldots, e_m \in \mathbb{N}$ and let $\mathbf{m} = \prod_{i=1}^m \mathbf{x_i}^{e_i}$ be a monomial of $\mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_m}]$. Following the descent described in Section 3.1, we have

$$\mathcal{A}(\mathbb{F}_{2^n}) \ni (\mathbf{mf})_V = \sum_{k=1}^{n} [(\mathbf{mf})_\mathbf{V}]_k^\downarrow \theta_k, \text{ with } [(\mathbf{mf})_\mathbf{V}]_k^\downarrow \in \mathcal{A}(\mathbb{F}_{2^n}).$$

The equation $\mathbf{f} = \mathbf{0}$ clearly implies $\mathbf{mf} = \mathbf{0}$, hence $[(\mathbf{mf})_\mathbf{V}]_k^\downarrow = 0$ for $k = 1, \ldots, n$. We can then add these new equations to the polynomial system (2). The equations obtained in this way all share the same structure. More precisely, their minimal representatives, due to the normal form computation modulo $\mathcal{I}_{\text{fe}}$, are all *affine multi-linear* in $\mathbb{F}_{2^n}[y_{1,1}, \ldots, y_{m,n'}]$. Moreover, thanks to the evaluations done during the deployments, each *block of variables* $X_i = \{y_{i,1}, \ldots y_{i,n'}\}$ naturally corresponds to the variable x_i. From these structures, we deduce the following result.

Lemma 1. *Let $\mathbf{f} \in \mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_m}]$ be a multivariate polynomial with degree $< 2^t$ in each variable. Let $e_1, \ldots, e_m \in \mathbb{N}$ and $\mathbf{m} = \prod_{i=1}^m \mathbf{x_i}^{e_i}$ be a monomial of $\mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_m}]$. There exist polynomials $p_{j,k} \in \mathbb{F}_2[y_{1,1}, \ldots, y_{m,n'}]$ such that $[(\mathbf{mf})_\mathbf{V}]_k^\downarrow = \sum_{j=1}^n p_{j,k} [\mathbf{f_V}]_j^\downarrow$. Each polynomial $p_{j,k}$ has degree $\leq W(e_i)$ with respect to every block of variables $X_i = \{y_{i,1}, \ldots y_{i,n'}\}$, $1 \leq i \leq m$. Moreover, each minimal representative of $[(\mathbf{mf})_\mathbf{V}]_k^\downarrow$ has degree $\leq \max_{0 \leq e_i' < 2^t} W(e_i + e_i')$ w.r.t. each block of variables $X_i, 1 \leq i \leq m$.*

This lemma implies that the new equations (obtained from $\mathbf{mf}$) are algebraic combinations of the original ones (obtained from $\mathbf{f}$). In particular, they can *a priori* be recovered "in a hidden form" with any Gröbner basis algorithm at degree $D_{apriori} = mt + \sum_{j=1}^m W(e_j)$. The value $D_{apriori}$ is the degree that the equations $[(\mathbf{mf})_\mathbf{V}]_k^\downarrow$ should have *a priori* from the algebraic dependencies of

Lemma 1. It is the sum of the degree of the deployments of $\mathbf{f}$ (at most mt) and the degree of each polynomial $p_{j,k}$ $\left(\text{at most } \sum_{j=1}^{m} W(e_j)\right)$. However, Lemma 1 also implies that the $[(\mathbf{mf})_{\mathbf{V}}]_k^{\downarrow}$ only have degree $D_{actual} = \sum_{j=1}^{m}(\max_{0 \le e'_j < 2^t} W(e_j + e'_j))$. Thus:

$$D_{apriori} - mt \le D_{actual} \le D_{apriori}.$$

Therefore, $[(\mathbf{mf})_{\mathbf{V}}]_k^{\downarrow}$ may have a *degree drop* as large as mt depending on the monomial $\mathbf{m}$ chosen. The existence such low-degree relations compared generic systems makes Gröbner basis algorithms faster in practice and allows a linearization strategy.

Following the general method of Macaulay [35], we will linearize the polynomial system $\mathcal{S}_{\text{alg}} \cup \{\ldots,\ldots, [(\mathbf{mf})_{\mathbf{V}}]_1^{\downarrow}, [(\mathbf{mf})_{\mathbf{V}}]_2^{\downarrow}, \ldots, [(\mathbf{mf})_{\mathbf{V}}]_n^{\downarrow}, \ldots,\ldots\}$ using the low-degree equations identified in this section. The choice of the monomials $\mathbf{m}$ used to generate the equations are particularly important for the efficiency of the linearization strategy. In particular, the equations with the lowest degrees are the most interesting ones since they involve less monomial terms. Of course, this strategy requires that a substantial subset of all low-degree relations are linearly independent.

3.3 Linear Dependencies

In the previous section, we explained how low-degree relations can be produced. To be used in a linearization strategy, these equations must be linearly independent. In this section, we describe two sources of linear dependencies.

Frobenius Transforms. The first source of linear dependencies is due to the Frobenius endomorphism (as identified in [22]). Let $\{\theta_1, \ldots, \theta_n\}$ be a basis of $\mathbb{F}_{2^n}/\mathbb{F}_2$. The set $\{\theta_1^2, \ldots, \theta_n^2\}$ is another basis of $\mathbb{F}_{2^n}/\mathbb{F}_2$. Let $a_{ij} \in \mathbb{F}_2$ be such that $\theta_j^2 = \sum_i a_{ij}\theta_i$. We have $\mathbf{f_V}^2 = \sum_{j=1}^{n} \left[\mathbf{f_V}^2\right]_j^{\downarrow} \theta_j$. However, $\mathbf{f_V}^2 = \left(\sum_{j=1}^{n} [\mathbf{f_V}]_j^{\downarrow} \theta_j\right)^2 = \sum_{j=1}^{n} [\mathbf{f_V}]_j^{\downarrow} \theta_j^2 = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} a_{ij} [\mathbf{f_V}]_j^{\downarrow}\right) \theta_i$. Thus, we obtain $\left[\mathbf{f_V}^2\right]_i^{\downarrow} = \sum_{j=1}^{n} a_{ij} [\mathbf{f_V}]_j^{\downarrow}$. In other words, the polynomials $\left[\mathbf{f_V}^2\right]_1^{\downarrow}, \ldots, \left[\mathbf{f_V}^2\right]_n^{\downarrow}$ are linear combinations of $[\mathbf{f_V}]_1^{\downarrow}, \ldots, [\mathbf{f_V}]_n^{\downarrow}$. Decomposing $\mathbf{f_V}$ as a sum of monomials, we deduce that $\left[\mathbf{f_V}^2\right]_i^{\downarrow} = \sum_{\mathbf{m} \in \mathtt{Mon}(\mathbf{f_V})} [(\mathbf{mf})_{\mathbf{V}}]_i^{\downarrow} = \sum_{j=1}^{n} a_{ij} [\mathbf{f_V}]_j^{\downarrow}$. This provides a non trivial linear relation between some low-degree equations. More generally, we obtain similar relations if we replace $\mathbf{f}$ by $\mathbf{m}'\mathbf{f}$ in the above equation (for any monomial $\mathbf{m}'$). These linear dependencies can be easily detected and prevented. Indeed, we can simply avoid every monomial $\mathbf{m}$ that is the leading term of $(\mathbf{m}'\mathbf{f})$ for some $\mathbf{m}'$.

Vector Dependencies. Another source of dependency, that we call *vector dependencies*, is induced by the vector space V. To illustrate the phenomena, consider the simplest polynomial $\mathbf{g} = \mathbf{x_1} \in \mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_m}]$. We have $\mathbf{g_V} = \nu_1 y_{1,1} + \nu_2 y_{1,2} + \cdots + \nu_{n'} y_{1,n'} \in \mathcal{A}(\mathbb{F}_{2^n})$. More generally, $(\mathbf{g^{2^i}})_{\mathbf{V}} = \nu_1^{2^i} y_{1,1} +$

$\nu_2^{2^i} y_{1,2} + \cdot + \nu_{n'}^{2^i} y_{1,n'} \in \mathcal{A}(\mathbb{F}_{2^n})$. Since $y_{1,1}, \ldots, y_{1,n'}$ are linearly independent, we obtain for any $k > n'$ a non trivial linear dependency by considering different g^{2^i}'s, i.e. $\exists \beta_1, \ldots, \beta_k \in \mathbb{F}_{2^n} \setminus \{(0, \ldots, 0)\}$ such that $\beta_1 (\mathbf{g^{2^{i_1}}})_{\mathbf{V}} + \beta_2 (\mathbf{g^{2^{i_2}}})_{\mathbf{V}} + \cdots + \beta_k (\mathbf{g^{2^{i_k}}})_{\mathbf{V}} = 0$. This simple example can be easily generalized to $\mathbf{g} = \mathbf{mf}$ with $\mathbf{m}$ a monomial. Such linear dependency can be clearly prevented during the generation of equations $[(\mathbf{mf})_V]_i^{\downarrow}$'s by considering monomials $\mathbf{m} = \prod_{i=1}^m \mathbf{x_i}^{e_i}$, with $0 \leq e_i < 2^{n'} < 2^n$.

3.4 Description of the Linearization Algorithm

For any positive integer d, let MLinMonB(d) be the set of multi-linear monomials in $\mathbb{F}_2[y_{1,1}, \ldots, y_{m,n'}]$ of degrees $\leq d$ with respect to each block $X_i = \{y_{i,1}, \ldots, y_{i,n'}\}$. The image of MLinMonB($d$) in $\mathcal{A}(\mathbb{F}_2)$ is a basis of the vector subspace $\mathcal{A}(\mathbb{F}_2)_d$ composed of elements in $\mathcal{A}(\mathbb{F}_2)$ with a minimal representative having degrees $\leq d$ with respect to each block X_i. Let also Mon(d) be the set of monomials $\mathbf{m} = \prod_{i=1}^m \mathbf{x_i}^{e_i} \in \mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_n}]$ with $1 \leq e_i < 2^{n'}$, such that all $[(\mathbf{mf})_{\mathbf{V}}]_k^{\downarrow}$ $(1 \leq k \leq n)$ are in $\mathcal{A}(\mathbb{F}_2)_d$. Finally, let $E(d) := n \cdot \#\mathrm{Mon}(d)$ and $M(d) := \#\mathrm{MLinMonB}(d)$.

We are now ready to describe Algorithm 1, a simple linearization algorithm for solving Problem 2. The algorithm constructs a sub-matrix of the Macaulay (see Definition 2) matrix $\mathcal{M}_d$ for system 2. We first gather $M(d)$ equations $[(\mathbf{mf})_{\mathbf{V}}]_i^{\downarrow}$ with $\mathbf{m} \in \mathrm{Mon}(d)$. By definition, all these equations are in $\mathcal{A}(\mathbb{F}_2)_d$. Hence, they can be decomposed with respect to the basis MLinMonB(d). We then form the corresponding linear system $\mathcal{S}_{lin}$ over $\mathbb{F}_2$, where each row corresponds to the coefficients involves in the equations and each column corresponds to an element in MLinMonB(d). We finally solve the linear system. This simple algorithm, that we call *Sub-Macaulay*, is not aimed to be optimal in practice but to derive complexity bounds. The general linearization strategy and our analysis below rely on a heuristic assumption formalized below:

Assumption 1. *With a probability exponentially close to one, the equations generated by Algorithm 1 are linearly independent.*

Particularly, the assumption states that the solutions of $\mathcal{S}_{lin}$ are in one-to-one correspondence with the solutions of Problem 2.

3.5 Complexity Bounds for Solving Problem 2

We now derive an upper bound on the complexity of Algorithm 1. The main task is to estimate the values of $M(d)$ (number of columns in $\mathcal{S}_{lin}$) and $E(d)$ (number of equations in $\mathcal{S}_{lin}$). Due to the field equations, we only have multi-linear monomials, i.e. variables can only have exponents 0 or 1. Therefore, the number of monomials of total degree $d', 0 \leq d' \leq d$ involving variables of the block X_i is $\binom{n'}{d'}$. For the m blocks, we get:

$$M(d) = \left(\sum_{d'=0}^{d} \binom{n'}{d'} \right)^m . \tag{3}$$

```
    Input: $\mathbf{f} \in \mathbb{F}_{2^n}[\mathbf{x_1}, \dots, \mathbf{x_m}]$ of degree in each variable is bounded by $2^t - 1$, and
           $V$ a $\mathbb{F}_2$-vector subspace $\subset \mathbb{F}_{2^n}$ of dimension $n'$.
    Result: If not empty, the finite set $\{s_1, \dots\} \subset V^m$ such that $\mathbf{f}(s_i) = \mathbf{0}$.
 1  begin
 2      Let $d$ be the smallest integer such that $E(d) \geq M(d)$;
 3      Mon $\leftarrow$ []; $\mathcal{S}_{\mathrm{alg}} \leftarrow$ []   // Empty lists of monomials and equations
 4      for $k = 1, \dots, \lceil E(d)/n \rceil$ do
 5          Randomly pick a monomial $\mathbf{m} \in \mathrm{Mon}(d) \setminus \mathrm{Mon}$;
 6          Let $[(\mathbf{mf})_{\mathbf{V}}]^{\downarrow}_{1}, \dots, [(\mathbf{mf})_{\mathbf{V}}]^{\downarrow}_{n} \in \mathbb{F}_2[y_{1\,1}, \dots, y_{m\,n_m}]$ be such that
            $(\mathbf{mf})_{\mathbf{V}} = \sum_{k=1}^{n} [(\mathbf{mf})_{\mathbf{V}}]^{\downarrow}_{k}\, \theta_k$;
 7          $\mathcal{S}_{\mathrm{alg}} \leftarrow \mathcal{S}_{\mathrm{alg}} \cup ([(\mathbf{mf})_{\mathbf{V}}]^{\downarrow}_{1}, \dots, [(\mathbf{mf})_{\mathbf{V}}]^{\downarrow}_{n})$; Mon $\leftarrow$ Mon $\cup$ $[\mathbf{m}]$;
 8      end
 9      Construct $\mathcal{S}_{lin}$ the linear system over $\mathbb{F}_2$ obtained by linearizing $\mathcal{S}_{\mathrm{alg}}$;
10      If $\mathcal{S}_{lin}$ has solutions then solve $\mathcal{S}_{lin}$ and return $\{s_1, \dots, \}$ else return no
        solution end;
11  end
```

Algorithm 1: Sub-Macaulay.

By definition, the degree of each variable $\mathbf{x_i}$ occurring in the monomials of $\mathbf{f}$ may have any degree between 0 and $2^t - 1$. By Lemma 1, the degree of $[(\mathbf{mf})_{\mathbf{V}}]^{\downarrow}_{k}$ with respect to X_i is $\max_{0 \leq e''_i < 2^t} W(e'_i + e''_i) = W\left(\left\lfloor \frac{e'_i}{2^t} \right\rfloor\right) + t$. Therefore, the number of exponents $e'_i, 1 \leq e'_i \leq 2^{n'}$ leading to degree d' with respect to the block X_i is $2^t \binom{n'-t}{d'-t}$. As a consequence[3]

$$E(d) := n\, 2^{tm} \left(\sum_{d'=t}^{d} \binom{n'-t}{d'-t} \right)^m . \tag{4}$$

We derive the following asymptotic bound (proven in Appendix A) on the minimal value d that allows linearization.

Lemma 2. *Let α be such that $1 - \alpha < 1/2 < \alpha < 1$. Assuming that $n' = n^\alpha, m = n^{1-\alpha}$ and $t = m - 1$, then $E(d) \geq M(d)$ for $d \approx \frac{n^\alpha}{2}$ when n is large enough.*

In the next table, we have computed the smallest d_{real} such that $E(d_{\mathrm{real}}) \geq M(d_{\mathrm{real}})$ for different values of n and α. Then, we compute $\beta_{\mathrm{real}} = \log_n(d_{\mathrm{real}})$. According to Lemma 2, the theoretical value predicted for $\beta = \log_n(d)$ is $\beta_{\mathrm{theo}} = \alpha - \frac{1}{\log(n)}$. The last column columns shows that β_{theo} is extremely close to β_{real}.

[3] Note that we assume that $d \geq t$.

n	d_{real}	α	β_{real}	$\left(\alpha - \frac{1}{\log(n)}\right) - \beta_{\text{real}}$
100000	1114	2/3	0.6093	-0.0029
1000000	5102	2/3	0.617956	-0.0014
10000000	23466	2/3	0.62435	-0.00069
100000000	108353	2/3	0.62936	-0.00032
1000000	1285	0.55	0.51815	-0.018
10000000	4331	0.55	0.51951	-0.012
100000000	14738	0.55	0.52105	-0.0089
1000000000	50577	0.55	0.52266	-0.0061
10000000000	174773	0.55	0.52425	-0.0043

Finally, we obtain an estimate on the complexity:

Theorem 2. *Let α, n', m, t be as in Lemma 2. Under Assumption 1, Problem 2 can asymptotically be solved in time $O(2^{\omega\tau})$ and memory $O(2^{2\tau})$, where ω is the linear algebra constant and $\tau \approx \frac{n}{2}$.*

Proof. The above linearization algorithm reduces the problem to linear algebra on a matrix of rank $M(d)$. According to Lemma 4 (Appendix A), we get that $\log(M(d)) \approx m \log \binom{n^\alpha}{n^\beta} \approx n^{(1-\alpha)}\, n^\beta(\alpha-\beta)\log(n) = n^{\left(1-\frac{1}{\log(n)}\right)} = \frac{n}{2}$. □

4 Application to ECDLP over Binary Fields

4.1 Diem's Variant of Index Calculus

Let E be an elliptic curve defined over $\mathbb{F}_{2^n}$ by the equation

$$E : y^2 + xy = x^3 + x^2 + \mathbf{a_6},\ \text{for some } \mathbf{a_6} \in \mathbb{F}_{2^n}. \tag{5}$$

Semaev' summation polynomials $\mathbf{S_r}$ [40] are multivariate polynomials with the following property: $\mathbf{S_r}(\mathbf{x_1}, \dots, \mathbf{x_r}) = \mathbf{0}$ for some $\mathbf{x_1}, \dots, \mathbf{x_r} \in \overline{\mathbb{F}_{2^n}}$ if and only if there exist $\mathbf{y_1}, \dots, \mathbf{y_r} \in \overline{\mathbb{F}_{2^n}}$ such that $(\mathbf{x_i}, \mathbf{y_i}) \in E(\overline{\mathbb{F}_{2^n}})$ and $(\mathbf{x_1}, \mathbf{y_1}) + \cdots + (\mathbf{x_r}, \mathbf{y_r}) = P_\infty$.

Proposition 1 ([40]). *The summation polynomials of the elliptic curve* (5) *are recursively given by:* $\mathbf{S_2}(\mathbf{x_1}, \mathbf{x_2}) := \mathbf{x_2} + \mathbf{x_1}$, $\mathbf{S_3}(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}) := \mathbf{x_1}^2\mathbf{x_2}^2 + \mathbf{x_1}^2\mathbf{x_3}^2 + \mathbf{x_1}\mathbf{x_2}\mathbf{x_3} + \mathbf{x_2}^2\mathbf{x_3}^2 + \mathbf{a_6}$ *and for $r \geq 4$ and any $k, 1 \leq k \leq r-3$, the r-th summation polynomial is*

$$\mathbf{S_r}(\mathbf{x_1}, \dots, \mathbf{x_r}) := Res_{\mathbf{X}}\big(\mathbf{S_{r-k}}(\mathbf{x_1}, \dots, \mathbf{x_{m-k-1}}, \mathbf{X}), \mathbf{S_{k+2}}(\mathbf{x_{r-k}}, \dots, \mathbf{x_r}, \mathbf{X})\big).$$

Moreover, the polynomial $\mathbf{S_r}$ is symmetric and has degree 2^{r-2} in each variable $\mathbf{x_i}$ as soon as $r \geq 2$. The cost of constructing this polynomial is bounded by $2^{O((r-1)^2)}$.

Following Diem [14], we use summation polynomials in the sieving stage of an index calculus algorithm. Let V be a vector subspace of $\mathbb{F}_{2^n}/\mathbb{F}_2$ with a dimension n' to be fixed later. We define the *factor basis* $\mathcal{F}_V$ as

$$\mathcal{F}_V := \{(\mathbf{x}, \mathbf{y}) \in E(\mathbb{F}_{2^n}) | \mathbf{x} \in V\}.$$

Since the abscissas of points $\in E$ are uniformly distributed in $\mathbb{F}_{q^n}$ [13,14], we can assume that the set $\mathcal{F}_V$ has size about $2^{n'}$. During the sieving stage, we compute about $2^{n'}$ relations $P_\infty = a_i P + b_i Q + \sum_{P_j \in \mathcal{F}_V} e^i_j P_j$ with $P_j \in \mathcal{F}_V$ for randomly chosen integer couples (a_i, b_i). Each relation is obtained by solving an instance of the following problem, for some integer parameter m to be fixed later.

Problem 3. Let a_i, b_i be fixed random integers and $R = a_i P + b_i Q$. Find - if any - $(\mathbf{x_1}, \dots, \mathbf{x_m}) \in V^m$ such that $\mathbf{S_{m+1}}(\mathbf{x_1}, \dots, \mathbf{x_m}, (R)_x) = \mathbf{0}$.

Clearly, this problem is a particular instance of Problem 2.

4.2 A Linearization Strategy for Solving ECDLP over $\mathbb{F}_{2^n}$

We now apply the analysis of Section 3 to Problem 3. Let $\alpha, 1/2 < \alpha < 1.$ be a parameter that will be optimized later. We set $n' := n^\alpha$ and $m := \ n^{1-\alpha}$ as in Lemma 2. According to Proposition 1, the $(m+1)$th Semaev's polynomial $\mathbf{S_{m+1}}$ can be computed in time $O(2^{t_1})$, where

$$t_1 \approx m^2 \approx n^{2(1-\alpha)}.$$

For each relation computed in the sieving stage, we generate and solve an instance of Problem 2 where $\mathbf{f}$ has degree 2^{m-1} with respect to each of the m variables. According to Theorem 2, each instance can be solved in time $O(2^{\omega\tau})$ and memory $O(2^{2\tau})$, where $\tau \approx n/2$. The probability that a point R can be written as a sum of m factor basis elements is $\frac{1}{m!}$ [14]. Hence, we need $m!2^{n'}$ trials on average to obtain $2^{n'}$ valid relations. Since $\log(m!) \approx n^{(1-\alpha)} \log n^{(1-\alpha)}$, the total cost of the sieving stage is bounded by $O(2^{t_2})$ where

$$t_2 \approx n^{(1-\alpha)} \log n^{(1-\alpha)} + n^\alpha + \omega\,\frac{n}{2}.$$

Finally, the last step of our algorithm consists in (sparse) linear algebra on a matrix of rank about $2^{n'}$ with elements of size about n bits. The computation time of this part can be approximated by $O(2^{t_3})$, where $t_3 \approx \omega' n^\alpha$, ω' being the sparse linear algebra constant. Finally, we obtain the following theorem.

Theorem 3. *Under Assumption 1, ECDLP over $\mathbb{F}_{2^n}$ can asymptotically be solved in time $O(2^{\omega\, t})$, where $t \approx n/2$.*

This has to be compared with the algorithm presented [14] which is was so far the best algebraic index calculus based approach. For $q = 2$ and n prime, the algorithm of [14] has complexity $e^{O\left(n \log(n)^{1/2}\right)}$ (but such complexity holds without any heuristic assumption). The complexity of our approach is also very close to $O(2^{n/2})$; the complexity of generic algorithms (e.g. Pollard's rho algorithm).

5 Experimental Results

5.1 Validation of the Heuristic Assumption

We have experimentally checked the validity of Assumption 1 using the computer algebra system MAGMA. During the sieving stage, most of polynomial systems generated have no solution (only $1/m!$ polynomial systems will produce a solution). Thus we mainly check the assumption for polynomial systems with *no solution* in the vector space V. In order to validate this assumption we proceeded in the following way. We chose many random polynomials $\mathbf{f}$ with degree $2^t < 2^{m-1}$ in each of its $m < 5$ variables. The coefficients are in $\mathbb{F}_{2^n}$ with n a prime less than 40. Then, for each of these polynomials, we construct a large binary matrix $\mathfrak{M}$ of size $N \times M$. This matrix represents the deployed polynomials $[(\mathbf{mf})_V]_1^{\downarrow}, \ldots, [(\mathbf{mf})_V]_n^{\downarrow}$ for all monomials in $\mathrm{Mon}(d)$ $\big($with d is the smallest integer with $M(d) < E(d)\big)$. We avoid the ones corresponding to possible linear dependencies as identified in Section 3.3. We want to demonstrate that a random square submatrix of size $M \times M$ of $\mathfrak{M}$ is full rank. We recall that the probability that a random $N \times M$ boolean matrix has rank r is $P(N, M, r) = 2^{-NM} \prod_{j=0}^{r-1} (2^N - 2^j) \prod_{j=0}^{r-1} (2^M - 2^j) / \prod_{j=0}^{r-1} (2^r - 2^j)$. Hence $P(100, 100, 100)$ is only 28.8% but $P(105, 100, 100) = 96.9\%$. For this reason, we consider submatrices $\mathfrak{M}'$ of $\mathfrak{M}$ of size $(M+5) \times M$. We check that the rank is M or $M-1$ or $M-2$; for a random Boolean matrix the probability is 99.999982%. We repeated the test 100 times and deduced an approximation of the success probability. In all our experiments we always obtain $\approx 100\%$ of success. In particular, we validated the assumption for Problem 3 using Semaev's summation polynomials with $m+1$ variables ($m = 2, \ldots, 4$). These validations represent a huge computational effort since some of the matrices $\mathfrak{M}$ encountered during the experiments had more than 10000 columns.

5.2 Gröbner Basis Computations

We performed actual Gröbner basis computation using the FGb software [20] to compare the theoretical number of operations with a more realistic value.

n	m	Number of Operations (GB)	Theoretical bound bound (Algo 1)
41	2	$2^{23.5}$	$M(d)^2 \approx 2^{60}$
67	2	$2^{37.1}$	$M(d)^2 \approx 2^{90}$
97	2	$2^{51.1}$	$M(d)^2 \approx 2^{125}$
131	2	$2^{74.5}$	$M(d)^2 \approx 2^{160}$

For instance, for $n = 131$, $m = 2$ we can solve the point decomposition problem in $2^{74.5}$ operations using a variant of the hybrid method [8]. We compare the cost of solving Semaev's equations using Algorithm 1 (complexity of solving a linear system of rank $M(d)$) and a Gröbner basis computation. This is the dominating part in the complexity of our index calculus based algorithm. Remark that the number of operations reported in this table are much below the theoretical estimate for Algorithm 1. We have a gain of (at least) a factor 2 in the exponent in favor of Gröbner computations. Further experiments suggest that a more advanced approach (i.e. Gröbner basis instead of linearization) can lead to an algorithm of better complexity. One can hope a gain of a factor m in the exponent. However, we are, for the moment, not able to provide theoretical evidences supporting such an improvement.

6 Conclusion and Perspectives

As a conclusion, we emphasize that the algorithm of Section 3 is of independent interest in the more general context of polynomial system solving. It shows that algebraic systems arising by deploying a multivariate polynomial equation from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$ are easier to solve than generic systems. We have underlined the intrinsic structures which help for solving such systems. This is an open problem how to use such structure in an F_4/F_5 based algorithm. We hope that such improvements may lead to a subexponential algorithm. Finally, the approach generalizes quite easily to other composite fields with "small" characteristics, resulting in similar algorithms with comparable asymptotic complexities. All these extensions will be discussed in an extended version. Finally, although the paper is mainly theoretical, we hope that it could be the building block towards the development of more efficient methods to solve the ECDLP problem.

Acknowledgements. We thank the reviewers of Eurocrypt 2012 for their very careful evaluation of our paper and their meaningful comments. We are also indebted to Sylvie Baudine for her comments on a preliminary version of this paper. The third author would like to thank the PolSys team for hosting him in the beginning of 2011, as well as Kristin Lauter and Daniele Micciancio since this paper was partially written in San Diego.

References

1. L. M. Adleman. A Subexponential Algorithm for the Discrete Logarithm Problem with Applications to Cryptography. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, SFCS '79, pages 55–60, Washington, DC, USA, 1979. IEEE Computer Society.
2. L. M. Adleman. The Function Field Sieve. In *Algorithmic number theory (Ithaca, NY, 1994)*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 108–121. Springer, Berlin, 1994.
3. L. M. Adleman, J. DeMarrais, and M. Huang. A Subexponential Algorithm for Discrete Logarithms over the Rational Subgroup of the Jacobians of Large Genus Hyperelliptic Curves over Finite Fields. In *Algorithmic number theory (Ithaca, NY, 1994)*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 28–40. Springer, Berlin, 1994.
4. L. M. Adleman and M. Huang. Function Field Sieve Method for Discrete Logarithms over Finite Fields. *Inform. and Comput.*, 151(1-2):5–16, 1999.
5. M. Bardet. *Étude des Systèmes Algébriques Surdéterminés. Applications aux codes Correcteurs et à la Cryptographie.* PhD thesis, Université Paris VI, 2004.

6. M. Bardet, J.-C. Faugère, and B. Salvy. Complexity of Gröbner Basis Computation for Semi-Regular Overdetermined Sequences over F_2 with Solutions in F_2. Technical Report 5049, INRIA, December 2003. Available at http://www.inria.fr/rrrt/rr-5049.html.
7. M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang. Asymptotic Expansion of the Degree of Regularity for Semi-Regular Systems of Equations. In P. Gianni, editor, *The Effective Methods in Algebraic Geometry Conference, Mega 2005*, pages 1 –14, May 2005.
8. L. Bettale, J.-C. Faugère, and L. Perret. Hybrid Approach for Solving Multivariate Systems over Finite Fields. *Journal of Math. Cryptology*, 3(3):177–197, 2010.
9. L. Bettale, J.-C. Faugère, and L. Perret. Cryptanalysis of HFE, multi-HFE and Variants for Odd and Even Characteristic. *Des. Codes Cryptography*, pages 1–46, 2012.
10. B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Universität Innsbruck, 1965.
11. H. Cohen and G. Frey, editors. *Handbook of Elliptic and Hyperelliptic Curve Cryptography.* Discrete Mathematics and its Applications. Chapman & Hall/CRC, 2005.
12. D. Coppersmith. Fast Evaluation of Logarithms in Fields of Characteristic Two. *IEEE Transactions on Information Theory*, 30(4):587–593, 1984.
13. C. Diem. On the Discrete Logarithm Problem in Elliptic Curves. *Compositio Mathematica*, 147:75–104, 2011.
14. C. Diem. On the Discrete Logarithm Problem in Elliptic Curves II. Presented at ECC 2011, 2011. http://www.math.uni-leipzig.de/ diem/preprints/dlp-ell-curves-II.pdf.
15. A. Enge and P. Gaudry. A General Framework for Subexponential Discrete Logarithm Algorithms. *Acta Arith.*, 102(1):83–103, 2002.
16. J.-C. Faugère. A New Efficient Algorithm for Computing Gröbner Basis (F4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999.
17. J.-C. Faugère. A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, ISSAC '02, pages 75–83, New York, NY, USA, 2002. ACM.
18. J-C. Faugère and A. Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems using Gröbner Bases. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 44–60. Springer, 2003.
19. J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich. Algebraic Cryptanalysis of McEliece Variants with Compact Keys. In *Eurocrypt 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 279–298. Springer Verlag, 2010.
20. J.-C. Faugère. FGb: A Library for Computing Gröbner Bases. In *Mathematical Software - ICMS 2010*, volume 6327 of *Lecture Notes in Computer Science*, pages 84–87. Springer Verlag, 2010.
21. J.-C. Faugère, F. Levy dit Vehel, and L. Perret. Cryptanalysis of MinRank. In *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 280–296, 2008.
22. J.-C. Faugère, L. Perret, C. Petit, and G. Renault. New Subexponential Algorithms for Factoring in $SL(2, \mathbb{F}_2^n)$. Preprint, 2011.
23. J.-C. Faugère, M. Safey El Din, P.-J. Spaenlehauer Computing Loci of Rank Defects of Linear Matrices using Gröbner Bases and Applications to Cryptology. In *ISSAC '10: Proceedings of the 2010 international symposium on Symbolic and*

algebraic computation, ISSAC '10, pages 257–264, New York, NY, USA, 2010. ACM. Best Student Paper Award.
24. J.-C. Faugère, M. Safey El Din, P.-J. Spaenlehauer. Gröbner Bases of Bihomogeneous Ideals Generated by Polynomials of Bidegree (1,1): Algorithms and Complexity. *Journal of Symbolic Computation*, 46(4):406–437, 2011.
25. J.-C. Faugère and S. Rahmany. Solving Systems of Polynomial Equations with Symmetries using SAGBI-Gröbner bases. In *ISSAC '09: Proceedings of the 2009 international symposium on Symbolic and algebraic computation*, ISSAC '09, pages 151–158, New York, NY, USA, 2009. ACM.
26. P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A Double Large Prime Variation for Small Genus Hyperelliptic Index Calculus. *Math. Comp.*, 76(257):475–492 (electronic), 2007.
27. P. Gaudry. An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves. In *Advances in cryptology—EUROCRYPT 2000 (Bruges)*, volume 1807 of *Lecture Notes in Comput. Sci.*, pages 19–34. Springer, Berlin, 2000.
28. P. Gaudry. Index Calculus for Abelian Varieties of Small Simension and the Elliptic Curve Discrete Logarithm Problem. *J. Symb. Comput.*, 44(12):1690–1702, 2009.
29. L. Granboulan, A. Joux, and J. Stern. Inverting HFE is Quasipolynomial. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 345–356. Springer, 2006.
30. A. Joux and R. Lercier. The Function Field Sieve in the Medium Prime Case. In *Advances in cryptology—EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Comput. Sci.*, pages 254–270. Springer, Berlin, 2006.
31. A. Kipnis and A. Shamir. Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 1999.
32. N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.
33. M. Kraitchik. *Théorie des Nombres.* Gauthier–Villards, 1922.
34. D. Lazard. Gröbner-Bases, Gaussian Elimination and Resolution of Systems of Algebraic Equations. In J. A. van Hulzen, editor, *EUROCAL*, volume 162 of *Lecture Notes in Computer Science*, pages 146–156. Springer, 1983.
35. F.S. Macaulay. *The Algebraic Theory of Modular Systems.*, volume xxxi of *Cambridge Mathematical Library.* Cambridge University Press, 1916.
36. F.S. Macaulay. Some Properties of Enumeration in the Theory of Modular Systems. *Proc. London Math. Soc.*, 26:531–55, 1927.
37. V. S. Miller. Use of Elliptic Curves in Cryptography. In Hugh C. Williams, editor, *CRYPTO*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1985.
38. J. Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In *EUROCRYPT*, pages 33–48, 1996.
39. J.M. Rojas. Solving Degenerate Sparse Polynomial Systems Faster. *J. Symbolic Computation*, 28:155–186, 1999.
40. I. Semaev. Summation Polynomials and the Discrete Logarithm Problem on Elliptic Curves. Available at http://eprint.iacr.org/2004/031.pdf, 2004.

A Proof of Lemma 2

In order to show Lemma 2, We use the following well-known technical result (a proof is given in [22] for instance).

Lemma 3. *Let n be an integer and let $\delta, 0 < \delta < 1/2$ be a number such that $\delta n \in \mathbb{N}$. Finally, let $\nu := \frac{\delta}{1-\delta}$. Then, the following inequalities hold:*

$$\binom{n}{\delta n} < \sum_{i=0}^{\delta n} \binom{n}{i} < \frac{1}{1-\nu} \binom{n}{\delta n}.$$

We can now come back to the proof of Lemma 2.

Proof (of Lemma 2). We recall that $1 - \alpha < \frac{1}{2} < \alpha < 1$, and $n' = n^{\alpha}, m = n^{(1-\alpha)}, t = m-1$. The number of equations in the system generated in Algorithm 1 is $E(d) = n\,2^{t\,m} \left(\sum_{d'=t}^{d} \binom{n'-t}{d'-t}\right)^m$ and the number of columns is $M(d) = \left(\sum_{d'=0}^{d} \binom{n'}{d'}\right)^m$. We try to find $\beta, 1/2 < \beta < \alpha < 1$ such that $E(d) \geq M(d)$, where $d = n^{\beta}$ for n big enough.

A sum of binomials can be bounded from above by the last binomial occurring in the sum. We then get:

$$E(d)^{\frac{1}{m}} = n^{\frac{1}{m}}\, 2^t \sum_{d'=t}^{d} \binom{n'-t}{d'-t} \geq n^{\frac{1}{m}}\, 2^t \binom{n'-t}{d-t}.$$

Now, let $\delta := \frac{d}{n^{\alpha}}$ and $\nu := \frac{\delta}{1-\delta}$. When n is large enough, we have $\delta < \frac{1}{3}$. This leads to $\nu \leq \frac{1}{2}$ and $\frac{1}{1-\nu} \leq 2$. We are now in position to apply Lemma 3.

$$M(d)^{\frac{1}{m}} \leq 2 \binom{n'}{d}. \tag{6}$$

Hence, we want to find d such that $n^{\frac{1}{m}}\, 2^t \binom{n'-d}{d-t} \geq 2\binom{n'}{d}$. We search now for the equality and consider the logarithm of each side of (6):

$$\log(n^{\frac{1}{m}}) + t + \log\binom{n'-t}{d-t} = 1 + \log\binom{n'}{d}. \tag{7}$$

The following result is useful to derive an asymptotical equivalent of this equality:

Lemma 4. *Let $\gamma < 1/2 < \beta < \alpha < 1$. For n large enough, we have:* $\log\binom{n^{\alpha}-n^{\gamma}}{n^{\beta}-n^{\gamma}} \approx (n^{\beta} - n^{\gamma})(\alpha - \beta)\log(n)$.

Proof. We have $\log\binom{n^{\alpha}-n^{\gamma}}{n^{\beta}-n^{\gamma}} \approx (n^{\alpha} - n^{\gamma})\,\log(n^{\alpha} - n^{\gamma}) - (n^{\beta} - n^{\gamma})\,\log(n^{\beta} - n^{\gamma}) - (n^{\alpha} - n^{\beta})\,\log(n^{\alpha} - n^{\beta})$ and thus

$$\begin{aligned}\log\binom{n^{\alpha}-n^{\gamma}}{n^{\beta}-n^{\gamma}} &\approx (n^{\alpha} - n^{\gamma})\,\alpha\,\log(n) - (n^{\beta} - n^{\gamma})\,\beta\,\log(n) - (n^{\alpha} - n^{\beta})\,\alpha\,\log(n)\\ &= -n^{\gamma}\,\alpha\,\log(n) - (n^{\beta} - n^{\gamma})\,\beta\,\log(n) + n^{\beta}\,\alpha\,\log(n) = (n^{\beta} - n^{\gamma})\,(\alpha - \beta)\,\log(n)\end{aligned}$$

Taking $\gamma = 0$, we deduce $\log\binom{n^{\alpha}}{n^{\beta}} \approx n^{\beta}(\alpha - \beta)\log(n)$. Then, using $\gamma = 1 - \alpha$ we get $\log\binom{n^{\alpha}-n^{1-\alpha}}{n^{\beta}-n^{1-\alpha}} \approx (n^{\beta} - n^{1-\alpha})(\alpha - \beta)\log(n)$. As a consequence (7) yields:

$$\begin{gathered} n^{\alpha-1}\log n + n^{1-\alpha} + (n^{\beta} - n^{1-\alpha})(\alpha - \beta)\log(n) = \log(2) + n^{\beta}(\alpha - \beta)\log(n)\\ n^{\alpha-1}\log n + n^{1-\alpha} - n^{1-\alpha}(\alpha - \beta)\log(n) = \log(2) \text{ and } n^{1-\alpha} \approx n^{1-\alpha}(\alpha - \beta)\log(n)\end{gathered}$$

So that $1 \approx (\alpha-\beta)\log(n)$. Thus, $\beta \approx \alpha - 1/\log(n)$ and $d^{\beta} \approx n^{\left(\alpha - \frac{1}{\log(n)}\right)} = n^{\alpha}/2$. This concludes the proof of Lemma 2. $\square$