On Polynomial Systems Arising from a Weil Descent

Christophe Petit* and Jean-Jacques Quisquater**

UCL Crypto Group,
Université catholique de Louvain
Place du Levant 3
1348 Louvain-la-Neuve (Belgium)
christophe.petit@uclouvain.be, jjq@uclouvain.be

Abstract. In the last two decades, many computational problems arising in cryptography have been successfully reduced to various systems of polynomial equations. In this paper, we derive new complexity bounds for a class of polynomial systems introduced by Faugère, Perret, Petit and Renault. Our analysis is based on the background heuristic assumption that Gröbner basis algorithms terminate shortly after the first non trivial degree fall occurs in the computation. We support this assumption in our case by new experimental data for small parameters. Interestingly, our analysis generalizes previous work on HFE cryptosystem. We then revisit the applications of these systems to the elliptic curve discrete logarithm problem (ECDLP) for binary curves, to the factorization problem in $SL(2, \mathbb{F}_{2^n})$ and to other discrete logarithm problems. As a main consequence, we provide a heuristic analysis showing that Diem's variant of index calculus for ECDLP requires a *subexponential* number of bit operations $O(2^{c\, n^{2/3} \log n})$ over the binary field $\mathbb{F}_{2^n}$, where c is a constant smaller than 2. According to our estimations, generic discrete logarithm methods are outperformed for any $n > N$ where $N \approx 2000$, but elliptic curves of currently recommended key sizes ($n \approx 160$) are not immediately threatened. The analysis can be easily generalized to other fields with "small" characteristic.

1 Introduction

While linear systems of equations can be efficiently solved with Gaussian elimination, polynomial systems are much harder to solve in general. After their introduction by Buchberger [15], Gröbner bases have become the most popular way to solve polynomial systems of equations, in particular after the developement of fast algorithms like F_4 [27] and F_5 [28]. Polynomial systems arising in cryptography tend to have a special structure that simplifies their resolution. In the last twenty years, many cryptographic challenges have been first reduced to polynomial systems of equations and then solved with fast and sometimes *dedicated* Gröbner basis algorithms [52,46,32,41,13,23,24,34,14,33].

Our contribution

In this paper, we revisit a particular class of polynomial systems introduced by Faugère et al. [36,37], together with their cryptographic applications. These systems naturally arise by deploying a multivariate polynomial equation over an extension field into a system of polynomial equations over the ground prime field (a technique commonly called *Weil descent*). As pointed out in [36,37], the block structure (called *multi-homogeneous*

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

structure in [37]) of the resulting equations may already lead to substantial complexity improvements in practice.

Our first contribution is a new complexity analysis of these systems. We first observe that the *low degree equations* identified in [36,37] provide very strong bounds on the *first fall degrees* of the systems. For many classes of polynomial systems, experimental and theoretical evidence show that the first fall degree provides a very good estimation of the *degree of regularity*, which in turn determines the complexity of Gröbner basis algorithms. Based on new experimental data for "small" parameters, we then explicitely *assume* that the same is true for polynomial systems arising from a Weil descent and we immediately deduce new (heuristic) complexity bounds for their resolution. Interestingly, we observe that our bounds naturally generalize previous bounds obtained for HFE systems.

We then apply our analysis to an elliptic curve discrete logarithm algorithm of Diem [22] in the case of binary fields [37]. After adapting and verifying our heuristic assumption in this particular setting, we show (under the assumption) that the elliptic curve discrete logarithm problem can be solved over the binary field $\mathbb{F}_{2^n}$ in *subexponential time*

$$O(2^{c\, n^{2/3} \log n}),$$

where c is a constant smaller than 2. For n prime, this problem was previously thought to have complexity $O(2^{n/2})$.

Finally, we discuss further applications of polynomial systems arising from a Weil descent, including the factorization problem in $SL(2, \mathbb{F}_{2^n})$, HFE and other discrete logarithm problems.

The great and various applications of polynomial systems arising from a Weil descent make our analysis particularly useful to cryptography. Although we focus on characteristic 2 in this paper, most of our results can be easily extended to other "small" characteristics.

Outline

The remaining of this paper is organized as follows. Section 2 contains most of the notations and definitions used in the paper. Section 3 provides general background on algebraic cryptanalysis with Gröbner basis. Section 4 contains our new analysis of polynomial systems arising from a Weil descent. The application to Diem's algorithm is detailed in Section 5 and other applications are discussed in Section 6. Finally, Section 7 concludes the paper.

2 Definitions and notations

We mostly follow the notations introduced in [36]. For any "small" prime p and any $n \in \mathbb{Z}$, we write $\mathbb{F}_{p^n}$ for the finite field with p^n elements. We see the field $\mathbb{F}_{p^n}$ as an n-dimensional vector space over $\mathbb{F}_p$ and we let $\{\theta_1, \ldots, \theta_n\}$ be a basis for $\mathbb{F}_{p^n}/\mathbb{F}_p$. With some abuses of notations, we use bold letters for all elements, variables and polynomials over $\mathbb{F}_{p^n}$ and normal letters for all elements, variables and polynomials over $\mathbb{F}_p$.

If $x_1, \ldots, x_N$ are variables defined over a field $\mathbb{K}$, we write $R := \mathbb{K}[x_1, \ldots, x_N]$ for the ring of polynomials in these variables. Given a set of polynomials $f_1, \ldots, f_\ell \in R$,

the *ideal* $I(f_1, \ldots, f_\ell) \subset R$ is the set of polynomials $\sum_{i=1}^{\ell} g_i f_i$, where, $g_1, \ldots, g_\ell \in R$. We write $\mathrm{Res}_{x_i}(f_1, f_2)$ for the *resultant* of $f_1, f_2 \in R$ with respect to the variable x_i. A *monomial* of R is a power product $\prod_{i=1}^{k} x_i^{e_i}$ where $e_i \in \mathbb{N}$. A *monomial ordering* for R is an ordering $>$ such that $m_1 > m_2 \Rightarrow m_1 m_3 > m_2 m_3$ for any monomials m_1, m_2, m_3 and $m > 1$ for any monomial m. The *leading monomial* $LM(f)$ of a polynomial $f \in R$ for a given ordering is equal to its largest monomial according to the ordering. Its *leading term* is the corresponding term. For any polynomial $f \in R$, we denote the set of monomials of f by $\mathtt{Mon}(f)$.

We measure the memory and time complexities of algorithms by respectively the number of bits and bit operations required. 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 $n > N \Rightarrow f(n) \leq cg(n)$. Similarly, we write o for the "small o" notation: given two functions f and g of n, we say that $f = o(g)$ if for any $\epsilon > 0$, there exists $N \in \mathbb{Z}$ such that for any $n > N$, we have $|f(n)| \leq \epsilon |g(n)|$.

Finally, we write ω for the *linear algebra constant*. Depending on the algorithm used for linear algebra, we have $2.376 \leq \omega \leq 3$.

3 Background on polynomial system resolution

Whereas linear systems can be solved with Gaussian elimination, polynomial systems of equations are usually solved with Gröbner bases.

3.1 Gröbner basis algorithms

Let R be a polynomial ring and let $>$ be a fixed monomial ordering for this ring. A *Gröbner basis* [15,20] of an ideal $I(f_1, \ldots f_\ell) \subset R$ is a basis $\{f'_1, \ldots, f'_{\ell'}\}$ of this ideal such that for any $f \in I(f_1, \ldots f_\ell)$, there exists $i \in \{1, \ldots, \ell'\}$ such that $\mathrm{LT}(f'_i) | \mathrm{LT}(f)$.

The first Gröbner basis algorithm was provided by Buchberger in his PhD thesis [15]. Lazard [48] later observed that computating a Gröbner basis is essentially equivalent to performing linear algebra on *Macaulay matrices* at a certain degree.

Definition 1 (Macaulay Matrix [49,50]). *Let R be a polynomial ring over a field K and let $\mathcal{B}_d := \{m_1 > m_2 > \cdots\}$ be the sorted set of all monomials of degree $\leq d$ for a fixed monomial ordering. Let $F := \{f_1, \ldots, f_\ell\} \subset R$ be a set of polynomials of degrees $\leq d$. For any $f_i \in F$ and $t_j \in \mathcal{B}_d$ such that $\deg(f_i) + \deg(t_j) \leq d$, let $g_{i,j} := t_j f_i$ and let $c_{i,j}^k \in K$ be such that $g_{i,j} = \sum_{m_k \in \mathcal{B}} c_{i,j}^k m_k$. The* Macaulay matrix *$\mathcal{M}_d(F)$ of degree d is a matrix containing all the coefficients $c_{i,j}^k$, such that each row corresponds to one polynomial $g_{i,j}$ and each column to one monomial $m_k \in \mathcal{B}_d$.*

The idea behind Lazard's observation is *linearization*: new equations for the ideal are constructed by algebraic combinations of the original equations, every monomial term appearing in the new equations is treated as an independent new variable, and the system is solved with linear algebra. Gröbner basis algorithms like F_4 [27] and F_5 [28] successively construct Macaulay matrices of increasing sizes and remove linear dependencies in the rows until a Gröbner basis is found. Moreover, they optimize the computation by avoiding monomials t_j that would produce trivial linear combinations such as $f_1 f_2 - f_2 f_1 = 0$. The complexity of this strategy is determined by the cost of linear algebra on the largest Macaulay matrix occuring in the computation.

3.2 Degree of regularity and first fall degree

The degree of the largest Macaulay matrix appearing in a Gröbner basis computation with the algorithm F5 is called the *degree of regularity* D_{reg}. For a "generic" sequence of polynomials $f_1, \ldots, f_\ell \in R$ (with $\ell \leq n$), this degree is equal to $1 + \sum_{i=1}^{\ell}(\deg(f_i) - 1)$ [7]. The degree of regularity can be precisely estimated in the case of *regular* and *semi-regular* sequences [7,10] and (assuming a variant of Fröberg conjecture) in a few other cases [30,12]. However, precisely estimating this value for other classes of systems (in particular for the various structured systems appearing in cryptanalysis problems) may be a very difficult task.

In practice, the degree of regularity may often be approximated by the first degree at which a non trivial *degree fall* occurs during a Gröbner basis computation.

Definition 2. *Let R be a polynomial ring over a field K. Let $F := \{f_1, \ldots, f_\ell\} \subset R$ be a set of polynomials of degrees $\leq D_{firstfall}$. The* first fall degree *of F is the smallest degree $D_{firstfall}$ such that there exist polynomials $g_i \in R$ with $\deg(f_i) + \deg(g_i) \leq D_{firstfall}$, satisfying $\deg(\sum_{i=1}^{\ell} g_i f_i) < D_{firstfall}$ but $\sum_{i=1}^{\ell} g_i f_i \neq 0$.*

We have $D_{reg} \geq D_{firstfall}$. Experimental and theoretical evidences have shown in various contexts that the two definitions often lead to very close numbers. This can intuitively be explained by the observation that an extremely large number of relations with a degree fall occur at the degree $D_{firstfall}$ or the degree $D_{firstfall}+1$ in these contexts, and the low degree relations can in turn be combined to produce lower degree relations [41]. Although this is not true in general (counter-examples exist for HFE- [29]), it seems to be true for "random systems" and many "random" instances of "cryptanalysis systems" including HFE and its variants [32,41,25,23,24,12]. In fact, the *first fall degree* has even sometimes been called *degree of regularity* in the cryptography community [25,23,24].

The new analysis of Faugère et al.'s systems presented in this paper relies on the assumption that the two definitions also produce very close results for these particular systems.

3.3 Algebraic cryptanalysis

Many polynomial systems arising in cryptanalysis are very far from generic ones. In fact, their special structures often induce lower degrees of regularity, hence much better time complexities. Gröbner basis techniques have successfully attacked many cryptosystems, including HFE and its variants [52,46,32,41,13,23,24], the Isomorphism of Polynomials [34,14] and some McEliece variants [33]. In many cases, the resolution of these systems could be accelerated using *dedicated* Gröbner basis algorithms that exploited the particular structures. As was first pointed out in [36,37], this is also the case for polynomial systems arising from a Weil descent.

4 Polynomial systems arising from a Weil descent

Let n, n', m be positive integers and let V be a vector subspace of $\mathbb{F}_{2^n}/\mathbb{F}_2$ with dimension n'. Let $\mathbf{f} \in \mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_m}]$ be a multivariate polynomial with degrees bounded by $2^t - 1$ with respect to all variables. In [36,37], Faugère et al. considered the following problem:

$$\text{Find } \mathbf{x_i} \in V, i = 1, \ldots, m, \text{ such that } \mathbf{f}(\mathbf{x_1}, \ldots, \mathbf{x_m}) = \mathbf{0}. \tag{1}$$

The constraints $\mathbf{x_i} \in V, i = 1, \ldots, m$ are called *linear constraints*. From now on, we assume that $mn' \approx n$ such that Problem (1) has about one solution on average. We also assume $n' \geq t$.

The *multilinear* case ($t = 1$) was first considered in [36] and later extended in [37]. We observe that the *monovariate* case ($m = 1$) also appeared in the cryptanalysis of HFE and its variants [32,41,23,25,12]. In these contexts, the linear constraints are trivial ($V = \mathbb{F}_{2^n}$) and the polynomial $\mathbf{f}$ has a special shape (it deploys as quadratic equations over $\mathbb{F}_2$). Interestingly, we will see that the special form of $\mathbf{f}$ in HFE contexts has little influence on the complexity of Problem (1).

4.1 Previous work by Faugère et al. [36,37]

Following [36,37], we reduce Problem (1) to a system of polynomial equations. We fix a basis $\{\theta_1, \ldots, \theta_n\}$ of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$ and a basis $\{\mathbf{v_i} | i = 1, \ldots, n'\}$ of V over $\mathbb{F}_2$. We define $m \cdot n'$ variables x_{ij} over $\mathbb{F}_2$ such that $\mathbf{x_i} = \sum_{j=1}^{n'} x_{ij}\mathbf{v_j}$ and we group them into m *blocks of variables* $X_i := \{x_{ij} | j = 1, \ldots, n'\}$. By substituting each $\mathbf{x_i}$ in $\mathbf{f}$, decomposing in the basis $\{\theta_1, \ldots, \theta_n\}$ and reducing by the *field equations* $x_{ij}^2 - x_{ij} = 0$, we obtain

$$\mathbf{0} = \mathbf{f}(\mathbf{x_1}, \ldots, \mathbf{x_m}) = \mathbf{f}\left(\sum_{j=1}^{n'} x_{1j}\mathbf{v_j}, \ldots, \sum_{j=1}^{n'} x_{mj}\mathbf{v_j}\right) = [\mathbf{f}]_1^{\downarrow}\theta_1 + \ldots + [\mathbf{f}]_n^{\downarrow}\theta_n \qquad (2)$$

for some $[\mathbf{f}]_1^{\downarrow}, \ldots, [\mathbf{f}]_n^{\downarrow} \in \mathbb{F}_2[x_{11}, \ldots, x_{mn'}]$ that depend on $\mathbf{f}$ and on the vector subspace V. Problem (1) can therefore be reformulated as finding a solution to the (algebraic) system

$$[\mathbf{f}]_1^{\downarrow} = 0, \ldots, [\mathbf{f}]_n^{\downarrow} = 0. \qquad (3)$$

Due to the bounds on the degrees of $\mathbf{f}$, the degrees of all polynomials $[\mathbf{f}]_k^{\downarrow}$ are bounded by t with respect to all blocks of variables. The resolution of System (3) can therefore be greatly accelerated using *block-structured* Gröbner basis algorithms [31,36,37].

Another very important observation made in [36,37] is that the ideal generated by the equations of System (3) contains an abnormally high number of *low degree equations* compared to generic systems. As an example of these equations, let us write the monomial $\mathbf{x_1}$ as $\sum_{i=1}^{n} [\mathbf{x_1}]_i^{\downarrow}\theta_i$, where each polynomial $[\mathbf{x_1}]_i^{\downarrow} \in \mathbb{F}_2[x_{11}, \ldots, x_{1,n'}]$ has degree 0 or 1. Defining $a_{ijk} \in \mathbb{F}_2$ such that $\theta_i\theta_j = \sum_k a_{ijk}\theta_k$, we obtain

$$\mathbf{x_1}\mathbf{f} = \left(\sum_{i=1}^{n} [\mathbf{x_1}]_i^{\downarrow}\theta_i\right)\left(\sum_{j=1}^{n} [\mathbf{f}]_j^{\downarrow}\theta_j\right) = \sum_{i,j,k=1}^{n} a_{ijk}\,[\mathbf{x_1}]_i^{\downarrow}\,[\mathbf{f}]_j^{\downarrow}\,\theta_k.$$

Decomposing $\mathbf{x_1}\mathbf{f}$ according to the basis $\{\theta_1, \ldots, \theta_n\}$, we see that every polynomial $[\mathbf{x_1}\mathbf{f}]_k^{\downarrow}$ can be written (modulo the field equations) as an algebraic combination of the polynomials (3)

$$[\mathbf{x_1}\mathbf{f}]_k^{\downarrow} = \sum_{i,j=1}^{n} a_{ijk}\,[\mathbf{x_1}]_i^{\downarrow}\,[\mathbf{f}]_j^{\downarrow} = \sum_{j=1}^{n} p_{ik}(x_{11}, \ldots, x_{1,n'})\,[\mathbf{f}]_j^{\downarrow} \qquad (4)$$

where each polynomial $p_{ik}(x_{11}, \ldots, x_{1,n'}) := \sum_{i=1}^{n} a_{ijk}\,[\mathbf{x_1}]_i^{\downarrow}$ has degree 0 or 1. On the other hand, since the polynomial $\mathbf{x_1}\mathbf{f}$ has degree at most 2^t in the variable $\mathbf{x_1}$, the

degree of each polynomial $[\mathbf{x_1 f}]_i^{\downarrow}$ is still bounded by t with respect to every block of variables X_i and its total degree is mt (instead of $mt+1$ as expected from Equation (4)). Similarly deploying $\mathbf{mf}$ for various monomials $\mathbf{m} \in \mathbb{F}_{2^n}[\mathbf{x_1}, \ldots, \mathbf{x_{n'}}]$, many more low degree equations can be generated [36,37].

In [36,37], Faugère et al. mostly focused on the *block structure* to obtain efficiency improvements on the resolution of System (3). They also attempted to linearize System (3) with the low degree equations, but they could only obtain loose bounds on the complexity of Problem (1)[1].

4.2 New experiments and heuristic assumption

In this paper, we follow a heuristic methodology commonly used to analyze HFE equations, that consists in approximating the *degree of regularity* of a particular system by its *first fall degree*. In the more general setting of System (3), we formalize the assumption as follows:

Assumption 1. *For a random polynomial* $\mathbf{f}$ *and a random vector space* V*, the degree of regularity and the first fall degree of System* (3) *are approximately equal. More precisely, we have* $D_{reg} = D_{firstfall} + o(D_{firstfall})$ *with a high probability.*

The first fall degree of System (3) is easily deduced from the low degree equations.

Proposition 1. *The first fall degree of System* (3) *is at most* $mt + 1$.

Proof. By definition, the proof amounts to showing the existence of a polynomial $g \neq 0$ with degree at most mt that can be written as $g(x_{11}, \ldots, x_{mn'}) = \sum_{i=1}^{n} p_i(x_{11}, \ldots, x_{m,n'}) \, [\mathbf{f}]_i^{\downarrow}$ for some polynomials $p_i \in K[x_{11}, \ldots, x_{mn'}]$ of degree 1. In fact, Equation (4) shows that we can take $g := [\mathbf{x_1 f}]_k^{\downarrow}$ for any k.

To validate Assumption (1), we experimentally study the degree of regularity of System (3) for various parameters n, m, n', t. For every set of parameters, we generate a random vector space V of dimension n' and a random multivariate polynomial $\mathbf{f}(\mathbf{x_1}, \ldots, \mathbf{x_m})$ with degree bounded by $2^t - 1$ with respect to each variable. We then perform a Weil descent on this polynomial, we append the field equations to the system and we apply the Magma function *Groebner* to the result. We repeat each experiment three times. In Table 1, we report the maximal degree D reached during the computation, as obtained from the *Verbose* output of the Magma function.

For every parameter set, the maximal degree reached by Gröbner basis computations on System (3) is much lower than the value that a generic system of equations (or even a generic *binary* system of equations) with the same degrees will have [7,8,9]. In fact, the maximal degree observed is either equal to $mt + 1$ or shortly larger than this value for most parameter sets. When $t = n'$, this degree is sometimes even as small as mt, probably due to a degenerency in the degrees of the original equations.

These experimental results provide good evidence that Assumption (1) is true, at least for small parameters. An experimental verification for larger parameters quickly

[1] The analysis of [36] turned out to be wrong since it ignored some linear dependencies between the low degree equations. These *vector dependencies* were identified and the analysis was adapted in [37], but the new bounds obtained in [37] are very loose, in particular compared to the experimental degrees that we observe in Section 4.2.

Table 1: Maximal degree reached in Gröbner Basis experiments for generic polynomials

t	n	n'	m	D
1	6	3	2	3
1	6	3	2	3
1	6	3	2	3
1	6	2	3	3
1	6	2	3	3
1	6	2	3	3
1	8	4	2	3
1	8	4	2	3
1	8	4	2	3
1	12	6	2	3
1	12	6	2	3
1	12	6	2	3
1	12	4	3	4
1	12	4	3	4
1	12	4	3	4
1	12	3	4	5
1	12	3	4	6
1	12	3	4	6

t	n	n'	m	D
1	12	2	6	6
1	12	2	6	6
1	12	2	6	8
1	15	5	3	5
1	15	5	3	5
1	15	5	3	4
1	15	3	5	6
1	15	3	5	6
1	15	3	5	6
1	16	8	2	3
1	16	8	2	3
1	16	8	2	3
1	16	4	4	6
1	16	4	4	5
1	16	4	4	5
1	16	2	8	8
1	16	2	8	9
1	16	2	8	8

t	n	n'	m	D
1	18	9	2	3
1	18	9	2	3
1	18	9	2	4
1	18	6	3	5
1	18	6	3	5
1	18	6	3	4
1	18	3	6	7
1	18	3	6	7
1	18	3	6	7
1	20	10	2	3
1	20	10	2	3
1	20	10	2	3
1	20	5	4	5
1	20	5	4	7
1	20	5	4	7
1	20	4	5	6
1	20	4	5	6
1	20	4	5	6

t	n	n'	m	D
2	6	3	2	5
2	6	3	2	5
2	6	3	2	6
2	6	2	3	7
2	6	2	3	7
2	6	2	3	7
2	8	4	2	5
2	8	4	2	5
2	8	4	2	5
2	9	3	3	7
2	9	3	3	8
2	9	3	3	7
2	12	4	3	8
2	12	4	3	7
2	12	4	3	7
2	12	3	4	10
2	12	3	4	9
2	12	3	4	11

t	n	n'	m	D
2	15	5	3	7
2	15	5	3	7
2	15	5	3	7
2	16	8	2	5
2	16	8	2	5
2	16	8	2	5
3	6	3	2	6
3	6	3	2	6
3	6	3	2	6
3	12	6	2	7
3	12	6	2	8
3	12	6	2	8
3	12	4	3	11
3	12	4	3	9
3	12	4	3	10
3	12	3	4	12
3	12	3	4	13
3	12	3	4	12

t	n	n'	m	D
3	15	5	3	10
3	15	5	3	10
3	15	5	3	10
3	16	8	2	8
3	16	8	2	7
3	16	8	2	7
3	16	4	4	14
3	16	4	4	13
3	16	4	4	13
4	8	4	2	9
4	8	4	2	8
4	8	4	2	9
4	12	4	3	12
4	12	4	3	12
4	12	4	3	13
4	15	5	3	13
4	15	5	3	13
4	15	5	3	14

becomes a challenging computational task due to the high degrees of regularity involved. The similarity of System (3) with HFE systems (for which a similar assumption has been widely verified) provides further confidence on its validity.

4.3 Heuristic complexity bounds for Problem (1)

Provided that Assumption 1 holds, the complexity of Problem 1 simply follows from the cost of linear algebra.

Proposition 2. *Under Assumption 1, Problem 1 can be solved with* standard *Gröbner basis algorithms (like F5) in time* $O(n^{\omega D})$ *and memory* $O(n^{2D})$*, where* ω *is the linear algebra constant and* $D \approx mt$*.*

In the monovariate case, this estimation reduces to $D \approx t$ which perfectly matches known cryptanalysis results on HFE algebraic systems [32,41]. Interestingly, the special shape of HFE polynomials (they deploy to *quadratic* equations over $\mathbb{F}_2$) seems to have no impact on the degree of regularity (although further restrictions on the shape may have an impact as pointed out in [23]). In the multilinear case, the estimation provided by Proposition 2 becomes $D \approx m$ which matches to the experimental data of [36].

As observed in [36,37], the block structure of System (3) can be exploited to accelerate its resolution. According to our analysis, the maximal degree appearing in the computation is approximately equal to the initial degree of Equations (3) and can be naturally distributed among the m blocks. Therefore, a *dedicated* Gröbner basis algorithm can be designed to exploit the sparsity induced by the block structure and reduce the time and memory complexities of solving Problem (1).

Proposition 3. *Under Assumption 1, Problem 1 can be solved with* block *Gröbner basis algorithms in time* $O((n')^{\omega D})$ *and memory* $O((n')^{2D})$*, where* ω *is the linear algebra constant and* $D \approx mt$*.*

Additional heuristic methods like the hybrid approach [11] may lead to substantial complexity improvements in practice, as was described in [36] for the multilinear case.

To conclude this section, we remark that the analysis of HFE by Granboulan et al. [41] can be easily adapted to provide an alternative analysis of Problem (1). We now turn to the main application (so far) of this problem.

5 Index calculus for elliptic curves

As pointed out in [37], an instance of Problem (1) appears in the relation search step of an index calculus algorithm for elliptic curves proposed by Diem [22]. Given a cyclic (additive) group G, a generator P of this group and another element Q of G, the discrete logarithm problem asks for computing an integer k such that $Q = kP$. Groups typically used in cryptography include the multiplicative groups of finite fields and cyclic subgroups of the Jacobian groups of elliptic and hyperelliptic curves. Index calculus algorithms [47,26] with *subexponential* complexities have long been obtained for the multiplicative groups of finite fields [1,19,2,5,43] and more recently for the Jacobian groups of hyperelliptic curves [3,39,38]. On the other hand, the best algorithms for solving elliptic curve discrete logarithms remained generic algorithms until very recently.

In 2004, Semaev introduced his *summation polynomials* and identified their potential application to build index calculus algorithms on elliptic curves [56] over prime fields $\mathbb{F}_p$. These ideas were independently extended by Gaudry [40] and Diem [21] to elliptic curves over composite fields $\mathbb{F}_{p^n}$. Following this approach, Gaudry [40] and later Joux and Vitse [44,45] obtained index calculus algorithms running faster than generic algorithms for any p and any $n \geq 3$. On the other hand, Diem [21,22] identified some families of curves with a subexponential time index calculus algorithm by letting p and n grow simultaneously in an appropriate way. As far as was known at the moment, the two families of elliptic curves recommended by standards [51] (elliptic curves over prime fields $\mathbb{F}_p$ or over binary fields $\mathbb{F}_{2^n}$ with n prime) remained immune to these attacks. In 2012, Faugère et al. [37] observed that the computation of the relations in an algorithm of Diem for binary fields [22] could be reduced to special instances of Problem (1). Moreover, they pointed out that the special structure of Problem (1) could be used to accelerate its resolution. In this section, we follow [37] and apply our analysis of Problem (1) to the relation search step of Diem's algorithm.

5.1 Diem's variant of index calculus

Let K be a finite field and let E be an elliptic curve over K defined by the equation

$$E : y^2 + xy = x^3 + x^2 + \mathbf{a_6} \tag{5}$$

for some $\mathbf{a_6} \in \mathbb{F}_{2^n}$. Semaev's *summation polynomials* $\mathbf{S_r}$ are multivariate polynomials satisfying $\mathbf{S_r}(\mathbf{x_1}, \ldots, \mathbf{x_r}) = \mathbf{0}$ for some $\mathbf{x_1}, \ldots, \mathbf{x_r} \in \bar{K}$ if and only if there exist $\mathbf{y_1}, \ldots, \mathbf{y_r} \in \bar{K}$ such that $(\mathbf{x_i}, \mathbf{y_i}) \in E(\bar{K})$ and $(\mathbf{x_1}, \mathbf{y_1}) + \cdots + (\mathbf{x_r}, \mathbf{y_r}) = P_\infty$ [56]. The summation polynomials of the curve (5) can be recursively computed as $\mathbf{S_2}(\mathbf{x_1}, \mathbf{x_2}) := \mathbf{x_2} + \mathbf{x_1}$, $\mathbf{S_3}(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}) := \mathbf{x_1}^2\mathbf{x_2}^2 + \mathbf{x_1}^2\mathbf{x_3}^2 + \mathbf{x_1}\mathbf{x_2}\mathbf{x_3} + \mathbf{x_2}^2\mathbf{x_3}^2 + \mathbf{a_6}$ and for any $r \geq 4$, any $k, 1 \leq k \leq r-3$:

$$\mathbf{S_r}(\mathbf{x_1}, \ldots, \mathbf{x_r}) := \mathrm{Res}_{\mathbf{X}}\left(\mathbf{S_{r-k}}(\mathbf{x_1}, \ldots, \mathbf{x_{m-k-1}}, \mathbf{X}), \mathbf{S_{k+2}}(\mathbf{x_{r-k}}, \ldots, \mathbf{x_r}, \mathbf{X})\right) \tag{6}$$

For $r \geq 2$, the polynomial $\mathbf{S_r}$ is symmetric and has degree 2^{r-2} in every variable $\mathbf{x_i}$ [56].

Summation polynomials were used by Gaudry [40], Joux and Vitse [44] and Diem [21,22] to compute relations in index calculus algorithms for elliptic curves over composite fields. The following variant is due to Diem [22].

1. **Factor Basis definition.** Fix two integers $m, n' < n$ with $mn' \approx n$ and a vector space $V \subset \mathbb{F}_{2^n}/\mathbb{F}_2$ of dimension n'. Let $\mathcal{F}_V := \{(\mathbf{x}, \mathbf{y}) \in E(K) | \mathbf{x} \in V\}$ be the *factor basis*.
2. **Relation search.** Find about $2^{n'}$ *relations* $a_i P + b_i Q = \sum_{j=1}^{m} P_{ij}$ with $P_{ij} \in \mathcal{F}_V$. For each relation,
   (a) Compute $R_i := a_i P + b_i Q$ for random integers a_i, b_i.
   (b) Solve Semaev's polynomial $\mathbf{S_{m+1}}(\mathbf{x_1}, \ldots, \mathbf{x_m}, (R_i)_x)$ with the constraints $\mathbf{x_i} \in V$.
   (c) If there is no solution, go back to (a).
3. **Linear Algebra.** Perform linear algebra on the relations to recover the discrete logarithm value.

In previous works [40,21,22,44], a Weil descent was applied to Semaev's polynomials and the resulting systems were solved with resultants or Gröbner basis algorithms. In these works, the complexity of the relation search step was derived from the complexity of solving generic systems. However as pointed out in [36,37] and further demonstrated in Section 4 of the present paper, polynomial systems arising from a Weil descent are very far from generic ones.

5.2 A new complexity analysis

We now revisit Diem's algorithm [22] and its analysis by [37] according to our new analysis of Problem (1). Let n, m, n' be integer numbers. Before starting Diem's algorithm, the $(m+1)$th summation polynomial must be computed. Using Collins' evaluation/interpolation method [18] for the resultant of Equation (6), this can be done in time approximately 2^{t_1} where[2]

$$t_1 \approx m(m+1). \tag{7}$$

We then compute about $2^{n'}$ relations. To obtain these relations, we solve special instances of Problem (1) where

$$\mathbf{f}(\mathbf{x_1}, \ldots, \mathbf{x_m}) := \mathbf{S_{m+1}}(\mathbf{x_1}, \ldots, \mathbf{x_m}, (a_i P + b_i Q)_x)$$

has degree 2^{m-1} with respect to every variable. Since Semaev's polynomials are clearly not random ones, the analysis requires to modify Assumption (1) as follows:

Assumption 2. *Let E be an elliptic curve defined over the field $\mathbb{F}_{2^n}$. Let V be a random vector space of dimension n' over $\mathbb{F}_2$ and let R be a random point on the curve. Let $\mathbf{f} := \mathbf{S_{m+1}}(\mathbf{x_1}, \ldots, \mathbf{x_m}, R_x)$ and let $\mathcal{S}$ be the corresponding System (3). For this system, we have $D_{reg} = D_{firstfall} + o(D_{firstfall})$ with a high probability.*

[2] To compute $\mathbf{S_{m+1}}$, we apply Collins' algorithm on S_k where $k = \lceil \frac{m+3}{2} \rceil$. This polynomial has degree $2^{\lceil (m-1)/2 \rceil}$ in each variable. Following Collins, Theorem 9, we have $t_1 \leq 2(m+1)m/2 = m(m+1)$.

To validate this new assumption, we apply Diem's algorithm to a randomly chosen binary curve $E : y^2 + xy = x^3 + \mathbf{a_4}x^2 + \mathbf{a_6}$ defined over $\mathbb{F}_{2^n}$, where $n \in \{11, 17\}$. We first fix $m \in \{2, 3\}$ and $n' := \lceil n/m \rceil$. We then generate a random vector space V of dimension n' and a random point R on the curve. As in Section 4.2, we finally use the *Groebner* function of Magma to solve Semaev's equation $\mathbf{S_{m+1}}(\mathbf{x_1}, \ldots, \mathbf{x_m}, R_x) = \mathbf{0}$ with the linear constraints. We also do the same experiments with the Koblitz curve $E : y^2 + xy = x^3 + x^2 + 1$. The maximal degrees reached during the computation are reported in Tables 2 and 3. The extra column T in this table corresponds to the degree of the $(m+1)$th summation polynomial with respect to every variable.

Table 2: Maximal degree D reached in Gröbner Basis experiments for Semaev's polynomials, random curves

n	m	n'	T	t	D
11	2	6	2	2	3
11	2	6	2	2	4
11	2	6	2	2	4

n	m	n'	T	t	D
11	3	4	4	3	7
11	3	4	4	3	7
11	3	4	4	3	7

n	m	n'	T	t	D
17	2	9	2	2	4
17	2	9	2	2	4
17	2	9	2	2	4

n	m	n'	T	t	D
17	3	6	4	3	9
17	3	6	4	3	8
17	3	6	4	3	7

Table 3: Maximal degree D reached in Gröbner Basis experiments for Semaev's polynomials, $E : y^2 + xy = x^3 + x^2 + 1$

n	m	n'	T	t	D
11	2	6	2	2	4
11	2	6	2	2	4
11	2	6	2	2	4

n	m	n'	T	t	D
11	3	4	4	3	7
11	3	4	4	3	7
11	3	4	4	3	7

n	m	n'	T	t	D
17	2	9	2	2	4
17	2	9	2	2	4
17	2	9	2	2	4

n	m	n'	T	t	D
17	3	6	4	3	9
17	3	6	4	3	8
17	3	6	4	3	8

The results provide good evidence in favor of Assumption (2). In fact, the maximal degrees reached in the computations are in all cases even *below* the predictions of Proposition (1). This phenomenon is due to the sparsity of Semaev's polynomials and will be exploited in future work (in particular, the degree of $\mathbf{S_{m+1}}$ with respect to every variable is 2^{m-1} but bounded by $2^m - 1$ in the analysis of Section 4). From now on in the analysis, we ignore this difference and analyze Semaev's polynomials as the random polynomials of Section 4.

Under Assumption (2), Step 2(b) of Diem's algorithm can be solved using a dedicated Gröbner basis algorithm taking advantage of the block structure, in a time $(n')^{\omega D}$, where $D \approx (m^2 + 1)$ and ω is the linear algebra constant. Once the x components of a relation have been computed, the y components can be found by solving m quadratic equations and testing each possible combination of the solutions. This requires a time roughly 2^m, that can be neglected. On average, the probability that a point $R_i := a_i P + b_i Q$ can be written as a sum of m points from the factor basis is $\frac{2^{mn'-n}}{m!}$ [22]. Assuming $mn' \approx n$, the total cost of the relation search step can therefore be approximated by 2^{t_2}, where

$$t_2 \approx m \log m + n' + \omega(m^2 + 1) \log n'. \tag{8}$$

The last step of Diem's algorithm consists in (sparse) linear algebra on a matrix of rank about $2^{n'}$ with about m elements of size about n bits per row. This step takes a

time approximately equal to $mn2^{\omega' n'} = 2^{t_3}$, where

$$t_3 \approx \log m + \log n + \omega' n' \tag{9}$$

and ω' is the *sparse* linear algebra constant. If Assumption (2) holds and if $mn' \approx n$, the total time taken by Diem's algorithm can be estimated by $T := 2^{t_1} + 2^{t_2} + 2^{t_3}$, where t_1, t_2, t_3 are defined as above.

5.3 On the hardness of ECDLP in characteristic 2

We now use Formulas (7) to (9) to evaluate the hardness of the elliptic curve discrete logarithm problem over the field $\mathbb{F}_{2^n}$ for "small" values of n. In our estimations, we conservatively use $\omega = 3$ and $\omega' = 2$. We consider $n \in \{50, 100, 160, 200, 500, 10^3, 2 \cdot 10^3, 5 \cdot 10^3, 10^4, 2 \cdot 10^4, 5 \cdot 10^4, 10^5, 2 \cdot 10^5, 5 \cdot 10^5, 10^6\}$ and $m \in \{2, \ldots, n/2\}$. For every pair of values, we compute the values t_1, t_2 and t_3 with Equations (7), (8) and (9) respectively. Finally, we approximate the total running time of Diem's algorithm by $2^{t_{max}}$ where $t_{max} := \max(t_1, t_2, t_3)$. For every value of n, Table 4 presents the data corresponding to the value m for which t_{max} is minimal. We point out that the numbers obtained here have to be taken cautiously since they all rely on Assumption 2 and involve a few approximations.

Table 4: Complexity estimates for Diem's algorithm in characteristic 2

n	m	n'	t_1	t_2	t_3	t_{max}
50	2	25	6	97	57	97
100	2	50	6	137	108	137
160	2	80	6	177	168	177
200	2	100	6	202	209	209
500	3	167	12	393	344	393
1000	4	250	20	664	512	664
2000	4	500	20	965	1013	1013
5000	6	833	42	1926	1682	1926
10000	7	1429	56	3020	2873	3020
20000	9	2222	90	4986	4462	4986
50000	11	4545	132	9030	9110	9110
100000	14	7143	210	14762	14306	14762

According to our estimations, Diem's version of index calculus (together with a sparse Gröbner basis algorithm) beats generic algorithms for any $n \geq N$, where N is an integer close to 2000. An actual attack for current cryptographically recommended parameters ($n \approx 160$) seems to be out of reach today, but the numbers in [37] suggest that medium-size parameters could be reachable with additional Gröbner basis heuristics like the hybrid method [11]. This will be investigated in further work.

Letting n grow and fixing $n' := n^\alpha$ and $m := n^{1-\alpha}$ for a positive constant $\alpha < 1$, we obtain

$$\begin{aligned} t_1 &\approx n^{2(1-\alpha)}, \\ t_2 &\approx (1-\alpha)n^{1-\alpha}\log n + n^\alpha + \max\left(\alpha\omega n^{2(1-\alpha)}\log n, n^\alpha + 3\log n\right), \\ t_3 &\approx (2-\alpha)\log n + \omega' n^\alpha \end{aligned}$$

Taking $\alpha := 2/3$, the relation search dominates the complexity of the index calculus algorithm and we deduce the following result.

Proposition 4. *Under Assumption 2, the discrete logarithm problem over $\mathbb{F}_{2^n}$ can asymptotically be solved in time $O(2^{cn^{2/3}\log n})$, where $c := 2\omega/3$ and ω is the linear algebra constant.*

In particular if the Gaussian elimination algorithm is used for linear algebra, we have $w = 3$ and $c = 2$. We stress that Proposition 4 holds even when n is prime. Until now, the best complexity estimates obtained in that case corresponded to generic algorithms that run in time $2^{n/2}$.

6 Further applications of Problem (1)

6.1 Factoring elements in $SL(2, \mathbb{F}_{2^n})$

The factorization problem in a non Abelian (multiplicative) group G is the following one: given a set of generators $\mathcal{S} := \{s_1, \ldots, s_k\}$ for this group and an element $h \in G$, the problem asks for a decomposition $h = \prod_{i=1}^{N} s_{m_i}$ as a product of the generators. The preimage security of *Cayley hash functions*, an interesting family of cryptographic hash functions with natural parallelism, directly relies on this problem [57,17,53,55]. The problem becomes potentially hard when additional restrictions are put on the length N of the product. For a family of groups of increasing size, the standard computational assumption is that no product of polynomial length can be computed in polynomial time, the complexity parameter being the logarithm of the size of the groups. The mere existence of these products in general depends on a famous conjecture of Babai on the diameter of Cayley graphs [6,42].

Using a sequence of reductions introduced in [54], Faugère et al. [36] reduced the factorization problem in $SL(2, \mathbb{F}_{2^n})$ to a particular instance of Problem (1) with $t = 1$, where

$$\mathbf{f}(\mathbf{x_1}, ..., \mathbf{x_m}) := \begin{pmatrix} \mathbf{1} & \mathbf{1} \end{pmatrix} \left[\prod_{i=1}^{m} \begin{pmatrix} \mathbf{x}+\mathbf{x_i} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{pmatrix} \right] \begin{pmatrix} \mathbf{1} \\ \mathbf{1} \end{pmatrix} \tag{10}$$

for some $\mathbf{x} \in \mathbb{F}_{2^n}$. The first fall degree of the corresponding system is at most $m + 1$. We remark that the polynomial $\mathbf{f}$ is not totally random since $\mathbf{f}(\mathbf{x_1}, ..., \mathbf{x_m}) = \mathbf{f}(\mathbf{x_m}, ..., \mathbf{x_1})$, so Assumption 1 needs to be adapted to this case. The experimental data presented in [36] supports the corresponding assumption. Combining [54,36] and the analysis of Section (4), we easily deduce that for any $\mathcal{S}$ and any h, a polynomial length factorization of h can be computed in probabilistic subexponential time. Since our estimation of the degree of regularity is smaller than in [36], we can also derive new smaller complexity estimates for this problem.

6.2 HFE and other discrete logarithm problems

As we pointed out above, System (3) can also be seen as a generalization of HFE systems. These systems have been intensively studied in the literature [32,41,25,23,24,12], and the assumption corresponding to Assumption (1) has been widely verified in this case. The specificity of HFE polynomials with respect to "random" polynomials is that they deploy as quadratic polynomials over the prime field. Interestingly, the polynomial systems arising from "generic" HFE polynomials seem to have the same degree of regularity as if they arised from random polynomials with the same degrees. It is however known that further restrictions on $\mathbf{f}$ may lower the degree of regularity [23].

Besides ECDLP, factoring in $SL(2, \mathbb{F}_{2^n})$ and HFE, the analysis of this paper can be applied to analyze index calculus algorithms over a wide variety of groups, including the Jacobian of higher genus curves. (These additional applications had also been identified by Faugère et al. [37,35]). In particular, discrete logarithm problems in the field $\mathbb{F}_{2^n}$ can be reduced to an instance of Problem (1) and then solved in heuristic time $O(2^{\frac{1}{2}\omega n^{1/2}\log n})$. The complexity of this approach does not compete with Coppersmith's algorithm [19] but is comparable to Adleman's first index calculus algorithm [1].

7 Conclusion and perspectives

In this paper, we revisited the complexity of solving a class of polynomial systems previously considered by Faugère et al. [36,37]. These systems appear when a multivariate polynomial over an extension field is deployed via a Weil descent into a system of polynomial equations over the ground prime field. We observed that these systems can be seen as natural extensions of HFE systems. Under a heuristic assumption commonly taken in the cryptanalysis community (in particular in the cryptanalysis of HFE variants), we derive new bounds on their resolution. Our bounds nicely generalize previous bounds on HFE.

The most proeminent consequence of our analysis so far is to the elliptic curve discrete logarithm problem (ECDLP) over binary fields, an application previously identified in [37]. We showed that ECDLP can be solved in *heuristic subexponential* time $O(2^{c\, n^{2/3}\log n})$ over the binary field $\mathbb{F}_{2^n}$, where c is a constant smaller than 2. This complexity is obtained with an index calculus algorithm due to Diem [21] and a block-structured Gröbner basis algorithm. In practice, the resulting algorithm is faster than generic algorithms (previously thought to be the best algorithms for this problem) for any n larger than N, where N is an integer approximately equal to 2000. In particular, binary elliptic curves of currently recommended sizes ($n \approx 160$) are not immediately threatened.

Besides ECDLP in characteristic 2, the systems introduced in [36,37] have a wide range of applications. We briefly discussed the factorization problem in $SL(2, \mathbb{F}_{2^n})$, HFE systems and other discrete logarithm problems. We leave a refinement of our analysis to the particular polynomials appearing in these applications to further work, similarly to what was done in [23] for HFE in odd characteristic.

All our complexity estimates are based on the heuristic assumption that the degrees of regularity of polynomial systems arising from a Weil descent are only slightly larger than their first fall degrees. This assumption was experimentally verified for small parameters

in the three different settings considered in this paper. The resemblance of our equations with HFE systems, for which the assumption has been widely verified, provides further confidence on its validity. We leave to further work the adaptation of the most recent theoretical results on HFE [12] to all polynomial systems arising from a Weil descent.

To conclude this paper, we point out that most of our results generalize quite easily to other composite fields with "small" characteristics, resulting in comparable asymptotic complexities.

Acknowledgements We are indebted to Sylvie Baudine for her help in improving this paper. The experimental data presented here was obtained with the help of the PolSys-UPMC team computing infrastructure. We thank Jean-Charles Faugère for his permission to use it. We also thank Jean-Charles Faugère, Ludovic Perret and Guénaël Renault for their useful comments on a preliminary version of this paper. Finally, Christophe Petit would like to thank Daniel Augot for his hospitality since this paper was partially written at LIX.

References

1. Leonard M. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography (abstract). In *FOCS*, pages 55–60. IEEE, 1979.
2. Leonard M. Adleman. The function field sieve. In Adleman and Huang [4], pages 108–121.
3. Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh A. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the jacobians of large genus hyperelliptic curves over finite fields. In Adleman and Huang [4], pages 28–40.
4. Leonard M. Adleman and Ming-Deh A. Huang, editors. *Algorithmic Number Theory, First International Symposium, ANTS-I, Ithaca, NY, USA, May 6-9, 1994, Proceedings*, volume 877 of *Lecture Notes in Computer Science*. Springer, 1994.
5. Leonard M. Adleman and Ming-Deh A. Huang. Function field sieve method for discrete logarithms over finite fields. *Inf. Comput.*, 151(1-2):5–16, 1999.
6. László Babai and Ákos Seress. On the diameter of permutation groups. *European J. Combin.*, 13(4):231–243, 1992.
7. Magali Bardet. *Etude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Université Paris 6, 2004.
8. Magali Bardet, Jean-Charles Faugère, and Bruno 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`.
9. Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *International Conference on Polynomial System Solving - ICPSS*, pages 71 –75, Nov 2004.
10. Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. 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.
11. Luc Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2010.
12. Luc Bettale, Jean-Charles Faugère, and Ludovic Perret. Cryptanalysis of HFE, Multi-HFE and Variants for Odd and Even Characteristic. *Des. Codes Cryptography*, pages 1–42, 2012. accepted.
13. Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Cryptanalysis of multivariate and odd-characteristic hfe variants. In Catalano et al. [16], pages 441–458.
14. Charles Bouillaguet, Jean-Charles Faugère, Pierre-Alain Fouque, and Ludovic Perret. Practical cryptanalysis of the identification scheme based on the isomorphism of polynomial with one secret problem. In Catalano et al. [16], pages 473–493.
15. Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, Universitt Innsbruck, 1965.

16. Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors. *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, volume 6571 of *Lecture Notes in Computer Science*. Springer, 2011.
17. Denis Charles, Eyal Goren, and Kristin Lauter. Cryptographic hash functions from expander graphs. *J. Cryptology*, 22(1):93–113, 2009.
18. George Collins. The calculation of multivariate polynomial resultants. *Journal of the Association for Computing Machinery*, 18:515–522, 1971.
19. Don Coppersmith. Fast evaluation of logarithms in fields of characteristic two. 30(4):587–593, 1984.
20. David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Springer Verlag, Berlin, Heidelberg, New York, 1 edition, 1992.
21. Claus Diem. On the discrete logarithm problem in elliptic curves. *Compositio Mathematica*, 147:75–104, 2011.
22. Claus Diem. On the discrete logarithm problem in elliptic curves II. 2011. `http://www.math.uni-leipzig.de/~diem/preprints/dlp-ell-curves-II.pdf`.
23. Jintai Ding and Timothy J. Hodges. Inverting HFE systems is quasi-polynomial for all fields. In *CRYPTO*, pages 724–742, 2011.
24. Jintai Ding and Thorsten Kleinjung. Degree of regularity for HFE-. *IACR Cryptology ePrint Archive*, 2011:570, 2011.
25. Vivien Dubois and Nicolas Gama. The degree of regularity of hfe systems. In *ASIACRYPT*, pages 557–576, 2010.
26. Andreas Enge and Pierrick Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta Arith.*, 102(1):83–103, 2002.
27. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, June 1999.
28. Jean-Charles 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.
29. Jean-Charles Faugère. Personnal communication, 2012.
30. Jean-Charles Faugère, Mohab Safey El Din, and Pierre-Jean Spaenlehauer. Computing loci of rank defects of linear matrices using gröbner bases and applications to cryptology. In *ISSAC*, pages 257–264, 2010.
31. Jean-Charles Faugère, Mohab Safey El Din, and Pierre-Jean Spaenlehauer. Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree (1, 1): Algorithms and complexity. *J. Symb. Comput.*, 46(4):406–437, 2011.
32. Jean-Charles Faugère and Antoine 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.
33. Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic cryptanalysis of mceliece variants with compact keys. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 279–298. Springer, 2010.
34. Jean-Charles Faugère and Ludovic Perret. Polynomial equivalence problems: Algorithmic and theoretical aspects. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 30–47. Springer, 2006.
35. Jean-Charles Faugère, Ludovic Perret, Christophe Petit, and Guénaël Renault. Personal communication, 2011.
36. Jean-Charles Faugère, Ludovic Perret, Christophe Petit, and Guénaël Renault. New subexponential algorithms for factoring in $SL(2, \mathbb{F}_2^n)$. `http://eprint.iacr.org/2011/598`, 2011.
37. Jean-Charles Faugère, Ludovic Perret, Christophe Petit, and Guénaël Renault. Improving the complexity of index calculus algorithms in elliptic curves over binary fields. Accepted at EUROCRYPT2012, 2012.
38. 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.
39. Pierrick 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 Computer Science*, pages 19–34. Springer, Berlin, 2000.

40. Pierrick Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *J. Symb. Comput.*, 44(12):1690–1702, 2009.
41. Louis Granboulan, Antoine Joux, and Jacques Stern. Inverting HFE is quasipolynomial. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 345–356. Springer, 2006.
42. Harald Andrés Helfgott. Growth and generation in $sl_2(z/pz)$. *Ann. of Math. (2)*, 167 (2):601–623, 2008.
43. Antoine Joux and Reynald 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.
44. Antoine Joux and Vanessa Vitse. Elliptic Curve Discrete Logarithm Problem over Small Degree Extension Fields. Application to the static Diffie-Hellman problem on $E(\mathbb{F}_{q^5})$. Cryptology ePrint Archive, Report 2010/157, 2010. `http://eprint.iacr.org/`.
45. Antoine Joux and Vanessa Vitse. Cover and Decomposition Index Calculus on Elliptic Curves made practical. Application to a seemingly secure curve over F_{p^6}. Cryptology ePrint Archive, Report 2011/020, 2011. `http://eprint.iacr.org/`.
46. Aviad Kipnis and Adi 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.
47. Maurice Kraitchik. *Théorie des nombres*. Gauthier-Villars, 1922.
48. Daniel Lazard. Gröbner-bases, Gaussian elimination and resolution of systems of algebraic equations. In *Proceedings of the European Computer Algebra Conference on Computer Algebra*, volume 162 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, 1983. Springer Verlag.
49. F.S. Macaulay. *The algebraic theory of modular systems.*, volume xxxi of *Cambridge Mathematical Library*. Cambridge University Press, 1916.
50. F.S. Macaulay. Some properties of enumeration in the theory of modular systems. *Proc. London Math. Soc.*, 26:531–55, 1927.
51. National Institute of Standards and Technology. Digital signature standard. Federal Information Processing Standards 186-3, 2009.
52. Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *EUROCRYPT*, pages 33–48, 1996.
53. Christophe Petit. *On graph-based cryptographic hash functions*. PhD thesis, Universit catholique de Louvain, 2009.
54. Christophe Petit. Towards factoring in $SL(2, \mathbb{F}_{2^n})$. Preprint, 2011.
55. Christophe Petit and Jean-Jacques Quisquater. Rubik's for cryptographers. Available at `http://perso.uclouvain.be/christophe.petit/index.html`, 2010.
56. Igor Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. Available at `http://www.isg.rhul.ac.uk/~ppai034/_pub/papers/Semaev%20%28Feb%29.pdf`, 2004.
57. Jean-Pierre Tillich and Gilles Zémor. Hashing with SL_2. In Yvo Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 40–49. Springer, 1994.