Bounding HFE with SRA

Christophe Petit*

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

Abstract. The Hidden Field Equation cryptosystem (HFE) is a public key encryption scheme whose security relies on the hardness of solving a system of polynomial equations over the finite field $\mathbb{F}_2$. This scheme and its generalizations have attracted a lot of attention by the cryptographic community. It is known that HFE polynomial systems are much easier to solve than generic systems, and in fact the parameters proposed in the original HFE cryptosystem can be broken in practice using Gröbner basis algorithms. Several theoretical explanations have been provided for this property, but all of them have so far relied on some plausible conjectures or heuristic assumptions. In this paper, we provide a rigourous bound on the complexity of solving a general class of polynomial systems including HFE systems. Our proof connects the polynomials constructed by Gröbner basis algorithms to the partial computation results of the Successive Resultants Algorithm (SRA), a recently introduced algorithm for finding roots of polynomials over finite fields. Besides, we provide a variant of SRA that may be of independent interest. We believe that our approach could have further applications on similar systems that were recently introduced in connection to the elliptic curve discrete logarithm problem over small characteristic fields.

1 Introduction

The Hidden Field Equation (HFE) cryptosystem is a public key encryption scheme introduced by Patarin at Eurocrypt'96 [19]. The security of HFE relies on the (supposed) hardness of solving some systems of polynomial equations over the finite field $\mathbb{F}_2$, a problem that is known to be hard for generic systems. Although the original HFE scheme has now been practically broken, several of its generalizations and signature versions are still considered as secure today [6].The theoretical study of HFE therefore remains of considerable interest, both from a theoretical point of view and as a first step towards understanding the security of its generalizations.

Besides, recent works have unveiled important similarities between the polynomial systems appearing in HFE cryptanalysis and the more general class of *polynomial systems arising from a Weil descent*, appearing in index calculus attacks against the elliptic discrete logarithm problem (ECDLP) over small characteristic fields or the discrete logarithm problem over small characteristic fields, and in an algorithm to solve the factorization problem in the non-Abelian group $SL(2, 2^n)$ [21,13,14]. Any new insight on HFE polynomial systems is likely to have an impact on these important problems as well.

One of the most succesful cryptanalysis techniques against HFE cryptosystem has been the use of generic Gröbner basis algorithms. It has been experimentally observed that these algorithms perform much better on HFE systems than on generic systems [12]. More precisely, the *degree of regularity* (a key parameter to estimate the complexity of Gröbner basis algorithms) appeared to be much smaller for HFE systems than for generic systems with the same number of variables and the same degrees. This experimental observation has been explained in several ways [15,8,7,1,21], but

* Supported by an F.R.S.-FNRS postdoctoral research fellowship at Université catholique de Louvain, Louvain-la-Neuve.

so far all the explanations have relied on some heuristic assumption or unproven conjecture. These experimental results on HFE and some of these heuristic analyses have also been extended to the more general class of polynomial systems arising from a Weil descent, covering all the applications mentioned above [21].

1.1 Our Contributions

In this paper, we prove an upper bound $\lceil (p-1)(\log_p \deg f + 1) \rceil + 1$ for the degree of regularity of any system arising from the Weil descent of a *monovariate* polynomial f over $\mathbb{F}_{p^n}$. This proves a conjecture of Petit and Quisquater [21] in the monovariate case and it implies a similar upper bound for the degree of regularity of HFE systems. All our upper bounds on the degrees of regularity imply upper bounds on the complexity of solving the corresponding systems.

Our proofs do not involve any heuristic assumption or unproven conjecture. Starting from a polynomial system arising from a Weil descent, we first modify this system to obtain another system of the form implicitely solved by the *Successive Resultants Algorithm* (SRA), a recently introduced algorithm to compute the roots of a polynomial over $\mathbb{F}_{p^n}$ [20]. We then study the behaviour of Gröbner basis algorithms on the new system. We prove that the degree of regularity of the new system is smaller or equal to the degree of regularity of the original system, and we relate the polynomials that can be computed by Gröbner basis algorithms at degree $\lceil (p-1)(\log_p \deg f+1) \rceil +1$ on the new system to the polynomials that are computed by SRA. Quite naturally, the proof crucially relies on the way these particular systems are generated.

Besides closing a 10-year old conjecture on HFE [12], our proof brings new insights on polynomial systems arising from a Weil descent in general, which could eventually lead to theoretical or practical progress on other applications such as the elliptic curve discrete logarithm problem in small characteristic fields. The variant of SRA that we introduce computes the squarefree split part of a polynomial over $\mathbb{F}_{p^n}$, and may be of independent interest.

1.2 Outline

This paper is organized as follows. In Section 2, we recall background results on Gröbner basis algorithms (Section 2.1), polynomial systems arising from a Weil descent (Section 2.2) and the Successive Resultants Algorithm (Section 2.3). We then prove our main results in Section 3 (an outline of the proof is provided at the beginning of that section). We conclude the paper in Section 4.

2 Preliminaries

2.1 Gröbner basis algorithms

Let K be a field and let $R := K[x_1, \ldots, x_n]$ be a polynomial ring over K. For any $g \in K$, we write $LM(g, >)$ for the leading monomial of g with respect to the monomial ordering $>$, or simply $LM(g)$ when $>$ is clear from the context. For any $g_1, \ldots, g_\ell \in R$, we write $I(g_1, \ldots, g_\ell)$ for the ideal generated $g_1, \ldots, g_\ell$. We recall that a *Gröbner basis* [2,5] of an ideal $I(g_1, \ldots g_\ell) \subset R$ with respect to an ordering $>$ is a basis $\{g'_1, \ldots g'_{\ell'}\}$ of this ideal such that for any $g \in I(g_1, \ldots g_\ell)$, there exists $i \in \{1, \ldots, \ell'\}$ such that $\mathrm{LM}(g'_i) | \mathrm{LM}(g)$.

Let $g_i \subset R$, $i = 1, \ldots, m$ and let $\mathcal{S}$ be the corresponding system of polynomial equations $g_1(x_1, \ldots, x_n) = g_2(x_1, \ldots, x_n) = \ldots = g_n(x_1, \ldots, x_n) = 0$. By solving $\mathcal{S}$, we mean finding values $x_i \in K$ such that all equations are satisfied. One of the main tools to solve polynomial systems of equations are Gröbner basis algorithms. Indeed for a lexicographic ordering, a Gröbner basis is

made of a "triangular" set of equations $\{h_1, \ldots, h_{m'}\}$, such that h_i only depends on the variables $X_i \subseteq \{x_1, \ldots, x_n\}$ and $X_{m'} \subset X_{m'-1} \subset \ldots \subset X_1$. The system can then be solved one variable at the time, starting from x_n.

In practice, it is usually more efficient to first compute a Gröbner basis for a degree reverse lexicographic ordering, and then to convert this basis into a Gröbner basis for the lexicographic ordering using an algorithm such as FGLM [11]. When the number of solutions is small, the comparative cost of the second step is negligible. We focus on the first step in this paper, namely the computation of a degree reverse lexicographic ordering Gröbner basis.

The key idea behind all Gröbner basis algorithms is linearization. These algorithms systematically construct new polynomials

$$h_{ij} := m_i g_j$$

where m_i is a monomial in R and the degree of h_{ij} is bounded by some d. The coefficients of the polynomials h_{ij} are then encoded into a *Macaulay matrix*, one row per polynomial and one column per monomial term. Finally, linear algebra is performed on the rows of the matrix, with the goal of obtaining new polynomials (by construction, algebraic combinations of the original ones) with lower degrees. If d is large enough, this process will finish and produce a set of equations containing a Gröbner basis.

If K is a finite field $\mathbb{F}_q$ for some "small" q, the whole approach is applied to an augmented system including the *field equations* $x_i^q - x_i = 0$.

Fast Gröbner basis algorithms such as F4 or F5 [9,10] progressively increase the degree d. After generating all polynomials up to a certain degree d (except for obvious linear dependencies), they perform a partial linear algebra step at this degree. If new polynomials of lower degrees are obtained, these polynomials are in turn multiplied by monomials to obtain new polynomials at degre d, etc. If this process finishes at degree d without providing a system from which a Gröbner basis can be extracted, the degree is increased by one. New polynomials are then added to the system, etc.

A simple Gröbner basis algorithm following these lines is described in Algorithm 1. In comparison, better algorithms such as F4, F5 and XL2 [9,10,4] may include strategies to avoid repetitions and trivial linear combinations. We point out that some Gröbner basis algorithms do not create any new polynomials from the small degree polynomials found during the computation [18,3]. These linearization algorithms *à la Lazard* are conceptually simpler but also less efficient. In this paper, a *Gröbner basis algorithm* will always refer to an algorithm such as Algorithm 1 that does take advantage of low degree polynomials encountered during the computation. This is for example the case of F4, F5 and XL2.

Algorithm 1 Simple Gröbner basis algorithm for degree reverse lexicographic ordering

$S \leftarrow \{g_1, \ldots, g_n\}$
$d \leftarrow \max\{\deg g_i\}$
$r \leftarrow n$
repeat
    $S \leftarrow \{h_{ij} := m_j g_i | \quad \deg h_{ij} \leq d, \quad m_j \text{ is a monomial}, \quad g_i \in S\}$
    Linearize the set S, eliminating terms with the highest degrees first
    if $\#S = r$ **then**
        $d \leftarrow d + 1$
    else
        $r \leftarrow \#S$
    end if
until S contains a Gröbner basis

The cost of Gröbner basis algorithms is mostly determined by the cost of linear algebra on Macaulay matrices. This cost is $O\left(\binom{n+d-1}{k}^{\omega}\right)$ operations over K, where $\omega < 3$ is the linear algebra constant and d is the *degree of regularity* of the system, which is the largest degree reached during the computation. It can be approximated by $O(n^{\omega d})$ if d is small compared to n. For random systems of n polynomials in n variables with degrees $d_1, \ldots, d_n$, the degree of regularity is expected to be $d = \sum_{i=1}^{n}(d_i - 1) + 1$. However, polynomial systems with particular structures can have much lower degrees of regularity, and can therefore be much easier to solve in practice.

2.2 Polynomial systems arising from a Weil descent

Let now $K := \mathbb{F}_{p^n}$ where p is a "small" prime and n is a natural number. Let $R := K[x_1, \ldots, m]$. The field K is a vector space of dimension n over $\mathbb{F}_p$. Let $V \subset K$ be a subspace of dimension $n' \approx n/m$, and let $f \in R$ with degree bounded by $p^t - 1$ in each variable. The following problem was introduced by Faugère, Perret, Petit and Renault in [13,14]:

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

This problem has several applications in cryptography, to the factorization problem in $SL(2, \mathbb{F}_{2^n})$ [13], to HFE cryptosystem [19,21] and to index calculus algorithms for the discrete logarithm problem and the elliptic curve discrete logarithm problem over finite fields of small characteristic [14,21].

In [13,14,21], the above problem was reduced to a polynomial system over $\mathbb{F}_p$ via a *Weil descent* as follows. First, two bases $\{\theta_1, \ldots, \theta_n\}$ and $\{v_1, \ldots, v_{n'}\}$ are fixed for $\mathbb{F}_{p^n}$ and V over $\mathbb{F}_p$. Then, mn' variables $x_{i,j}$ over $\mathbb{F}_p$ are introduced such that $x_i = \sum_{j=1}^{n'} x_{ij} v_j$. To model the constraints $x_i \in V$, these expressions are used to substitute the variables x_i in f, and the resulting equation is decomposed with respect to the basis $\{\theta_1, \ldots, \theta_n\}$. We thereby obtain a polynomial system

$$[f]_1^{\downarrow}(x_1, \ldots, x_n) = 0, \ldots, [f]_n^{\downarrow}(x_1, \ldots, x_n) = 0 \tag{1}$$

of n equations in mn' variables over $\mathbb{F}_p$, with degrees bounded by mt [14,21].

A particular and modified version of these systems appears in the cryptanalysis of the Hidden Field Equation (HFE) cryptosystem [19]. In that context, $V := \mathbb{F}_{p^n}$ (the whole field), $m = 1$ (monovariate case) and f is a secret polynomial of the particular form

$$f(x) = \sum_{p^i + p^j \leq D} f_{ij} x^{p^i + p^j} + \sum_{p^i \leq D} f_i x^{p^i} + f_0$$

for some $f_0, f_i, f_{ij} \in \mathbb{F}_{p^n}$. The Weil descent of f is "hidden" by two secret linear bijections L_1 and L_2 to form a public set of polynomials g_j such that $g_1(x_1, \ldots, x_n) := L_2 \circ [f]_1^{\downarrow} \circ L_1(x_1, \ldots, x_n), \ldots, g_n(x_1, \ldots, x_n) := L_2 \circ [f]_n^{\downarrow} \circ L_1(x_1, \ldots, x_n)$. An attacker against HFE cryptosystem is given some ciphertext $(c_1, \ldots, c_n) \in (\mathbb{F}_p)^n$ and must find $x_i \in \mathbb{F}_p$ such that $c_j = g_j(x_1, \ldots, x_n)$ for all j. It has long been observed that the two secret linear transformations do not influence the efficiency of a Gröbner basis attack on this system [12,15].

Two decades of research on HFE cryptosystem and its variants have provided strong evidence that the degrees of regularity of HFE systems are $O(\log \deg f)$, implying that the Gröbner basis attack on HFE is quasipolynomial. However, the various "proofs" of this result all rely on some heuristic assumption,[1] like the close equality of the first fall degree and the degree of regularity [8,7,21], generic behaviour of some subsystem of HFE system [15] or a variant of Fröberg's conjecture [1].

[1] We stress that because of a terminology confusion due to the interaction of two different communities, some statements on the "degree of regularity" in the literature actually refer to the "first fall degree", hence they require a heuristic assumption to imply complexity bounds on the resolution of the systems (see [21]).

For generic polynomial systems arising from a Weil descent, the degree of regularity was conjectured to be close to $mt + 1$ when $p = 2$ by Petit and Quisquater [21]. For larger p values, a generalization of their analysis would approximate the degree of regularity by $(p-1)mt+1$. Both approximations rely on the close equality of the first fall degree and the degree of regularity, which was verified in practice for small parameters [21,17,22].

2.3 The Successive Resultants Algorithm (SRA)

For any basis $\{v_1, \ldots, v_n\}$ of $\mathbb{F}_{p^n}$ over $\mathbb{F}_p$, we can recursively define $n+1$ functions $L_0, L_1, \ldots, L_n$ from $\mathbb{F}_{p^n}$ to $\mathbb{F}_{p^n}$ such that $L_0(x) = x$ and $L_i(x) = \prod_{c \in \mathbb{F}_p} L_{i-1}(x - cv_1)$ for $i > 0$. It is easy to prove [20] for all i that the function L_i is linear, that it evaluates to zero at exactly all the elements of the vector space $\langle v_1, \ldots, v_i \rangle$ and that we have $L_i(x) = L_{i-1}(x)^p - a_i L_{i-1}(x)$ where $a_i := (L_{i-1}(v_i))^{p-1}$.

Let now f be a monovariate polynomial over $\mathbb{F}_{p^n}$. It is easy to check that solving the system

$$\begin{cases} f(x_1) = 0 \\ x_j^p - a_j x_j = x_{j+1} \quad j = 1, \ldots, n-1 \\ x_n^p - a_n x_n = 0 \end{cases} \tag{2}$$

is equivalent to solving the equation $f(x) = 0$ with $x \in \mathbb{F}_{p^n}$.

The Successive Resultants Algorithm (SRA) uses the quasi-diagonal structure of this system to efficiently solve f with resultants. Let $f^{(1)} := f$. In the first step of SRA, the polynomials

$$f^{(j+1)}(x_{j+1}) = \mathrm{Res}_{x_j}\left(f^{(j)}(x_j), x_{j+1} - (x_j^p - a_j x_j)\right)$$

are successively computed for $j = 1, \ldots, n-1$. It can be shown [20] that each polynomial $f^{(j)}$ has the same degree as f and only depends on the variable x_j. In the second step of SRA, the polynomial

$$g^{(n)}(x_n) := \gcd\left(f^{(n)}(x_n), x_n^p - a_n x_n\right).$$

and the set $S_n := \{\hat{x}_n | g^{(n)}(\hat{x}_n) = 0\}$ are first computed. Then for $j = n, \ldots, 2$, the polynomials

$$g^{(j-1)}(x_{j-1}) := \gcd\left(f^{(j-1)}(x_{j-1}), \hat{x}_n - (x_{j-1}^p - a_{j-1} x_{j-1})\right)$$

are computed for all $\hat{x}_j \in S_j$, all roots of all these polynomials are computed, and a new set S_{j-1} is formed that contains all these roots. It can be shown [20] that each polynomial $g^{(j-1)}(x_{j-1})$ is split with degree at most p, that each set S_j contains at most $\deg f$ elements, and that the set S_1 eventually contains all the solutions of the equation $f(x) = 0, x \in \mathbb{F}_{p^n}$.

3 Bounding HFE with SRA

We are now ready to prove our results. In Section 3.1, we provide a preliminary result on the behaviour of Gröbner basis algorithms on polynomial systems that include some "field equations". In Section 3.2, we bound the degree of regularity of System (1) by the degree of regularity of (a slight modification of) System 2. We then focus on that new system. In Section 3.3, we essentially prove that all the polynomials computed in the first step of SRA can also be computed by a Gröbner basis algorithm without increasing the degree above our upper bound. When the system has at most one solution, we then prove in Section 3.3 that this solution can also be computed by

a Gröbner basis algorithm without increasing the degree above our upper bound, essentially by following the second step of SRA. Finally in Section 3.5, we generalize our proof to any monovariate polynomial over $\mathbb{F}_{p^n}$, using a variant of SRA that we introduce.

We insist that our proofs only hold for Gröbner basis algorithms like Algorithm 1, taking advantage of low degree polynomials found to construct further polynomials (this is the case for F4 and F5). In particular, they do not apply to linearization Gröbner basis algorithms *à la Lazard* such as XL in its basic version [18,3].

3.1 Field Equations and Gröbner Basis Computations

Our proof involves polynomial systems including either the equations $\mathcal{R}_1 := \{x_i^p - x_i | i = 1, \ldots, n\}$, or the equations $\mathcal{R}_2 := \{y_i^p - y_{i+1} = 0 | i = 1, \ldots, n-1\} \cup \{y_n^p - y_1 = 0\}$, or the equations $\mathcal{R}_3 := \{x_j^p - a_j x_j = x_{j+1} | j = 1, \ldots, n-1\} \cup \{x_n^p - a_n x_n = 0\}$ where the a_i values are as in Section 2.3. With an abuse of terminology, we will call all these equations "field equations".

We define the map $\varphi : \mathbb{F}_{p^n}[y_1, \ldots, y_n] \to \mathbb{F}_{p^n}[y_1, \ldots, y_n]$ as the "reduction modulo the equations $y_i^p - y_{i+1} = 0$". More precisely for any $g \in \mathbb{F}_{p^n}[y_1, \ldots, y_n]$, $\varphi(g)$ is the normal form of g with respect to the ideal generated by the polynomials of $\mathcal{R}_2$, for a graded reverse lexicographic ordering of the monomials. We also define the map $\phi : \mathbb{F}_{p^n}[x_1, \ldots, x_n] \to \mathbb{F}_{p^n}[x_1, \ldots, x_n]$ as the "reduction modulo the equations $x_i - (x_{i-1}^p - a_{i-1}x_{i-1}) = 0$". More precisely for any $g \in \mathbb{F}_{p^n}[x_1, \ldots, x_n]$, $\phi(g)$ is the normal form of g with respect to the ideal generated by the polynomials pf $\mathcal{R}_3$, for a graded reverse lexicographic ordering of the monomials. Clearly, images of polynomials by either ϕ and φ have degrees at most $p-1$ with respect to all variables.

We first prove the following result:

Lemma 1. *Let f, h be monovariate polynomials over $\mathbb{F}_{p^n}$. Let $\mathcal{S} := \{\phi(f(x_1)) = 0\} \cup \mathcal{R}_3$. Starting from this system, the polynomial $\phi(h(x_1)f(x_1))$ can be computed by a Gröbner basis algorithm at degree at most $(p-1)\lceil \log_p(\deg h + \deg f)\rceil + 1$.*

Proof. It is sufficient to prove the lemma for all monomials h. We proceed by induction on $\deg h$. If $\deg h = 0$, there is nothing to prove. The polynomial $\phi(x_1^i f(x_1))$ has degree $(p-1)\lceil \log_p(i + \deg f)\rceil$. Let us assume by induction that it can be computed at degree $(p-1)\lceil \log_p(i + \deg f)\rceil + 1$. Then the polynomial $x_1\phi(x_1^i f(x_1))$ can be computed at the same degree. Finally, the polynomial $\phi(x_1^{i+1} f(x_1))$ can be obtained by "reducing $x_1\phi(x_1^i f(x_1))$ modulo the equations $x_i - (x_{i-1}^p - a_{i-1}x_{i-1}) = 0$", which corresponds to adding an algebraic combination of these equations:

$$\phi(x_1^{i+1} f(x_1)) = x_1\phi(x_1^i f(x_1)) + \sum_{j=1}^{n-1} h_i(x_1, \ldots, x_n)(x_j^p - a_j x_j - x_{j+1}) + h_n(x_1, \ldots, x_n)(x_n^p - a_n x_n)$$

for some polynomials h_i, h_n with degrees smaller or equal to $(p-1)\lceil \log_p(i + \deg f) - 1\rceil$. This operation can be "blindly" performed by a Gröbner basis algorithm without increasing the degree. This shows that $\phi(x_1^{i+1} f(x_1))$ can be computed at degree $(p-1)\lceil \log_p(i + \deg f)\rceil + 1 \leq (p-1)\lceil \log_p(i + 1 + \deg f)\rceil + 1$, and it concludes the proof.

Similarly, we have

Lemma 2. *Let f, h be monovariate polynomials over $\mathbb{F}_{p^n}$. Let $\mathcal{S} := \{\varphi(f(x_1)) = 0\} \cup \mathcal{R}_2$. Starting from this system, the polynomial $\varphi(h(x_1)f(x_1))$ can be computed by a Gröbner basis algorithm at degree at most $(p-1)\lceil \log_p(\deg h + \deg f)\rceil + 1$.*

3.2 From HFE to SRA Systems

We now argue that the degree of regularity of HFE systems can be bounded by the degree of regularity of (the image by ϕ of) System (2). We first define a notion of equivalent systems of equations.

Definition 1. *We say two polynomials systems* $\{f_i(x_1, \ldots, x_n) = 0 | i = 1, \ldots, m\}$ *and* $\{g_i(x_1, \ldots, x_n) = 0 | i = 1, \ldots, m\}$ *defined over a finite field* K *are equivalent over* K *if there exists two linear permutations* $L_1 : K^n \to K^n$ *and* $L_2 : K^m \to K^m$ *such that* $f_i(x_1, \ldots, x_n) = L_2 \circ (g_i) \circ L_1(x_1, \ldots, x_n)$.

Equivalent systems of equations have the same degree of regularity, simply because any algebraic combination of the equations of one system leads to a similar algebraic combination with the same degrees in the other system.

Lemma 3. *Let* f *be a polynomial defined over* $\mathbb{F}_{p^n}$ *and let* $[f]_i^{\downarrow}$ *be the polynomials resulting from its Weil descent for an arbitrary basis. Let* L_1 *and* L_2 *be two permutations of* $(\mathbb{F}_p)^n$. *Then any system*

$$\begin{cases} g_i(x_1, \ldots, x_n) := L_2 \circ ([f]_i^{\downarrow}) \circ L_1(x_1, \ldots, x_n) = c_i & i = 1, \ldots, n \\ x_i^p - x_i = 0 & i = 1, \ldots, n. \end{cases} \tag{3}$$

is equivalent over $\mathbb{F}_{p^n}$ *to a system with the form*

$$\begin{cases} \varphi(f^{p^{i-1}}(y_i)) = c^{p^{i-1}} & i = 1, \ldots, n \\ y_i^p - y_{i+1} = 0 & i = 1, \ldots, n-1 \\ y_n^p - y_1 = 0. \end{cases} \tag{4}$$

Proof. This follows from Granboulan et al. [15], Section 4. Starting from System 3, the two permutations L_1 and L_2 can first be eliminated to lead to an equivalent system

$$\begin{cases} [f]_i^{\downarrow}(x_1, \ldots, x_n) = d_i & i = 1, \ldots, n \\ x_i^p - x_i = 0 & i = 1, \ldots, n \end{cases} \tag{5}$$

where $d_i \in \mathbb{F}_p$. Moreover, up to two additional permutations of the variables and the equations, we can assume that the basis chosen for the Weil descent is a normal basis $\{\theta, \theta^p, \ldots, \theta^{p^{n-1}}\}$. We then apply a linear change of variables $y_i := \sum_{j=1}^n x_j \theta^{p^{i+j-1}}$ to this system. Note that "modulo the field equations", we have $y_1 = x$, $y_2 = x^p$, $y_3 = x^{p^2}$, etc. A linear change of equations $\sum_{j=1}^n \theta^{p^{i+j-1}}(x_j^p - x_j)$ leads to $y_n^p - y_1 = 0$ and to $y_i^p - y_{i+1} = 0$ for $i = 1, \ldots, n-1$. A final linear change of equations $f_i := \sum_{j=1}^n [f]_j^{\downarrow} \theta^{p^{i+j-1}}$ leads to the equations $\varphi(f)$, $\varphi(f^p)$, $\varphi(f^{p^2})$, etc. Note that all these transformations are invertible.

Lemma 4. *The degree of regularity of System* (4) *is bounded by the degree of regularity of the truncated system*

$$\begin{cases} \varphi(f(y_1)) = c \\ y_i^p - y_i = 0 \qquad i = 1, \ldots, n. \end{cases} \tag{6}$$

Conversely, the degree of regularity of System (6) *is bounded by the maximum of* $\lceil (p-1)(\log_p \deg f + 1) \rceil + 1$ *and the degree of regularity of System* (4).

Proof. We first note that the equations removed in System (6) are algebraically redundant in System (4). Indeed, the equation $f(y_2) = c^p$ follows from

$$f(y_1) - c = 0 \Rightarrow \left((f(y_1))^{p-1} + c\,(f(y_1))^{p-2} + c^2\,(f(y_1))^{p-3} + \ldots + c^{p-1} \right) (f(y_1) - c) = 0$$
$$\Rightarrow (f(y_1))^p - c^p = f^p(y_1^p) - c^p = f^p(y_2) - c^p = 0.$$

All the polynomials involved in these relations have degrees smaller than $p \deg(f)$. By Lemma 2, the same relations can be computed "modulo the field equations $y_i^p - y_{i-1}$" with polynomials of degree at most $\lceil (p-1)(\log_p \deg f + 1) \rceil + 1$.

Proceeding recursively with at most $n-1$ steps at degree $\lceil (p-1)(\log_p \deg f + 1) \rceil + 1$, (linear combinations of) all equations of System (4) will be recovered if we compute algebraic multiples of System (6) with a Gröbner basis algorithm.

Lemma 5. *System* (6) *is equivalent over* $\mathbb{F}_{p^n}$ *to the system*

$$\begin{cases} \phi(f(x_1)) = c \\ x_j^p - a_j x_j = x_{j+1} \quad j = 1, \ldots, n-1 \\ x_n^p - a_n x_n = 0 \end{cases} \tag{7}$$

where the a_i *values are as in Section 2.3.*

Proof. This follows from the linear change of variables $x_i := L_{i-1}(y_1)$ and from Lemma 1 in [20].

We deduce the following Proposition:

Proposition 1. *The degree of regularity of System* (3) *is not larger than the degree of regularity of System* (7)*.*

In the next sections, we will bound the degree of regularity of System (7).

3.3 SRA First Step

We now show that the images by ϕ of all the polynomials computed in the first step of SRA can also be computed "blindly" by Gröbner basis algorithms without increasing the degree above $\lceil (p-1)(\log_p \deg f + 1) \rceil + 1$.

Lemma 6. *Let* $f^{(i)}$ *be the polynomials computed by SRA. Starting from System* (7)*, all the polynomials* $\phi(f^{(i)})$ *can be computed by a Gröbner basis algorithm at degree at most* $\lceil (p-1)(\log_p \deg f + 1) \rceil + 1$*.*

Proof. [2] Since $f^{(2)}(x_2) = Res_{x_1}(f(x_1), x_2 - (x_1^p - a_1 x_1))$, there exist two polynomials $h(x_1, x_2)$ and $g(x_1, x_2)$ such that

$$f^{(2)}(x_2) = h(x_1, x_2) f(x_1) + g(x_1, x_2)(x_2 - (x_1^p - a_1 x_1)).$$

Defining $h_1(x_1) := h(x_1, x_1^p - a_1 x_1)$, we deduce

$$f^{(2)}(x_1^p - a_1 x_1) = h_1(x_1) f(x_1) \tag{8}$$

and $\deg h_1 = p \deg f^{(2)} - \deg f = (p-1) \deg f$. By Lemma 1, the polynomial $\phi(f^{(2)}(x_2))$ can therefore be computed by a Gröbner basis algorithm at degree at most $\lceil (p-1)(\log_p \deg f + 1) \rceil + 1$. The argument can then be repeated recursively. There exists a polynomial of degree $(p-1) \deg f^{(2)} = (p-1) \deg f$ such that multiplying this polynomial by $f^{(2)}(x_2)$ leads to $f^{(3)}(x_3)$, etc. After $n-1$ steps at degree at most $\lceil (p-1)(\log_p \deg f + 1) \rceil + 1$, the Gröbner basis algorithm will have computed all polynomials $\phi(f^{(i)}(x_i))$.

[2] Proof corrected with the help of Timothy Hodges.

3.4 SRA Second Step, Single Root Case

We now temporarily assume that the polynomial f has at most one root. The general case will be considered in the next section.

When f has at most one solution, the only operations occuring in the second step of SRA are gcd computations. We therefore have

Lemma 7. *Let $f^{(j)}, g^{(j)}$ be the polynomials computed by SRA. If the polynomial f has at most one solution, then $\phi(f^{(j)})$ and $g^{(j)}$ can be computed by Gröbner basis algorithms at degree $\lceil (p-1)(\log_p \deg f + 1)\rceil + 1$.*

Proof. We have already seen that a Gröbner basis algorithm can compute $\phi(f^{(n)})$ at degree $\lceil (p-1)(\log_p \deg f + 1)\rceil + 1$. Because it is "reduced modulo the equation $x_n^p - a_n x_n$", the polynomial $\phi(f^{(n)}(x_n))$ is a polynomial in x_n only, with degree at most $p-1$. By definition, we have

$$g^{(n)}(x_n) = \gcd(f^{(n)}(x_n), x_n^p - a_n x_n) = \gcd(\phi(f^{(n)}(x_n)), x_n^p - a_n x_n).$$

Following Euclide's algorithm for the computation of gcds, the polynomial $g^{(n)}(x_n)$ can be computed at degree p.

If f has exactly one solution, then $f^{(n)}$ has exactly one solution too, hence $g^{(n)}$ is a polynomial of degree 1 that we can assume monic: $g^{(n)}(x_n) := x_n - \hat{x}_n$ where $\hat{x}_n \in \mathbb{F}_{p^n}$ is the unique root of $f^{(n)}$. From this polynomial and the polynomial $x_n - (x_{n-1}^p - a_{n-1}x_{n-1})$, a Gröbner basis algorithm can compute a new polynomial $\hat{x}_n - (x_{n-1}^p - a_{n-1}x_{n-1})$. Similarly by adding an appropriate algebraic multiple of $g^{(n)}(x_n)$ to the already computed $\phi(f^{(n-1)}(x_{n-1}))$, a Gröbner basis algorithm will obtain a new polynomial

$$\hat{f}_{n-1}(x_{n-1}) = \phi(f^{(n-1)}(x_{n-1})) \bmod (x_n - \hat{x}_n)$$

of degree at most $p-1$ in x_{n-1}. By definition, we have

$$\begin{aligned} g^{(n-1)}(x_{n-1}) &= \gcd\left(f^{(n-1)}(x_{n-1}), \hat{x}_n - (x_{n-1}^p - a_{n-1}x_{n-1})\right) \\ &= \gcd\left(\hat{f}^{(n-1)}(x_{n-1}), \hat{x}_n - (x_{n-1}^p - a_{n-1}x_{n-1})\right), \end{aligned}$$

which for the same reasons as above has degree 1 and can be computed at degree p. All the operations so far can be done by Gröbner basis algorithms without increasing the degree above the degree of the polynomials that were already computed. Proceeding recursively, we see that a Gröbner basis algorithm can recompute all the polynomials $g^{(i)}$ without increasing the degree above $\lceil (p-1)(\log_p \deg f + 1)\rceil + 1$.

If f has no solution, then $f^{(n)}$ has no solution either, hence $g^{(n)} = 1$. The other $g^{(i)}$ are all equal to 1, and can equally be computed without increasing the degree above $\lceil (p-1)(\log_p \deg f + 1)\rceil + 1$.

When f has no root, Gröbner basis algorithms will find the polynomial 1 and return the Gröbner basis $\{1\}$. When f has exactly one root, then all $g^{(i)}(x_i)$ are linear polynomials, hence they form a Gröbner basis of System (7) for any ordering. Therefore if the polynomial f has at most one root, a Gröbner basis algorithm will complete its computation without increasing the degree above $\lceil (p-1)(\log_p \deg f + 1)\rceil + 1$.

3.5 Degree of Regularity in the General Case

When f has more than one root, SRA's second step may include some factorization steps of some polynomials $g^{(i)}$, whereas Gröbner basis algorithms do not include partial factorizations as subroutines. To analyze the behaviour of Gröbner basis algorithms when f has several roots, we first introduce a variant of SRA. This variant does not include any factorization routine, but on the other hand it only returns the splitting part of f (removing all non linear factors) without explicitly returning the roots.

The first step of our variant is identical to SRA. In the second step, we first compute

$$\hat{g}^{(n)}(x_n) := \gcd\left(f^{(n)}(x_n), x_n^p - a_n x_n\right)$$

in the same way as $g^{(n)}$ is computed in SRA. However instead of factoring $g^{(n)}$ at this point, we successively compute $\hat{g}^{(j)}$ for $j = n-1, \ldots, 1$ as

$$\hat{g}^{(j-1)}(x_{j-1}) := \gcd\left(f^{(j-1)}(x_{j-1}), \hat{g}^{(j)}(x_{j-1}^p - a_{j-1}x_{j-1})\right).$$

By the properties of resultants and gcds, the factorization of each polynomial $\hat{g}^j$ only contains linear terms corresponding to the solutions of $f^{(j)}(x_i) = 0$. In particular, $\hat{g}^{(1)}$ is the squarefree split part of f.

Lemma 8. *Let $f^{(j)}, \hat{g}^{(j)}$ be the polynomials computed by our variant of SRA. The polynomials $\phi(f^{(j)})$ and $\phi(\hat{g}^{(j)})$ can be computed by Gröbner basis algorithms at degree $\lceil (p-1)(\log_p \deg f + 1)\rceil + 1$.*

Proof. By Lemma 6, a Gröbner basis algorithm can compute $\phi(f^{(n)})$ at degree $\lceil (p-1)(\log_p \deg f + 1)\rceil + 1$. We also showed that the polynomial $\phi(\hat{g}^{(n)}(x_n))$ can be computed at the same degree in the proof of Lemma 7. By elementary division, there exist polynomials $r_1^{(n-1)}$, $q_1^{(n-1)}$ with $\deg r_1^{(n-1)} < p \deg \hat{g}^{(n)}$ such that

$$r_i^{(n-1)}(x_{n-1}) = f^{(n-1)}(x_{n-1}) + q_1^{(n-1)}(x_{n-1})\hat{g}^{(n)}(x_{n-1}^p - a_{n-1}x_{n-1}).$$

By Lemma 1, the polynomial $\phi\left(q_i^{(n-1)}(x_{n-1})\hat{g}^{(n)}(x_{n-1}^p - a_{n-1}x_{n-1})\right) = \phi\left(q_i^{(n-1)}(x_{n-1})\hat{g}^{(n)}(x_n)\right)$ can be computed at the same degree, hence so can the polynomial $\phi\left(r_i^{(n-1)}(x_{n-1})\right)$. Similarly by following the Euclidean algorithm, a Gröbner basis algorithm can eventually compute $\phi(\hat{g}^{(n-1)}(x_{n-1}))$ without increasing the degree. All the polynomials $\phi(\hat{g}^{(j)})$ can then be recovered similarly.

We remark that the knowledge of $\phi(\hat{g}^{(1)})$ essentially gives us a Gröbner basis of System (7) for a lexicographical ordering with $x_n > x_{n-1} > \ldots > x_1$. Indeed this basis is trivially given by

$$\{\hat{g}^{(1)}(x_1)\} \cup \{x_j^p - a_j x_j - x_{j+1} | j = 1, \ldots, n-1\} \cup \{x_n^p - a_n x_n\}.$$

We now prove that a degree reverse lexicographic oredring Gröbner basis can also be computed at the degree $\lceil (p-1)(\log_p \deg f + 1)\rceil + 1$ (irrespectively of the particular degree ordering chosen).

Proposition 2. *Let f be a monovariate polynomial over $\mathbb{F}_{p^n}$. The degree of regularity of System (7) is bounded by $\lceil (p-1)(\log_p \deg f + 1)\rceil + 1$.*

Proof. Let $D := \lceil (p-1)(\log_p \deg f + 1) \rceil + 1$ and let I be the ideal generated by the equations of System (7). We show that any polynomial of degree less than or equal to D in I can be computed at degree D.

For convenience, let us define

$$\begin{cases} s_{j,e_j} := \phi\left(x_j^{e_j} \hat{g}^{(j)}(x_j)\right) & 0 \leq e_j < (p \deg \hat{g}^{(j+1)} - \deg \hat{g}^{(j)}), 1 \leq j < n, \\ s_{n,e_n} := \phi\left(x_n^{e_n} \hat{g}^{(n)}(x_n)\right) & 0 \leq e_n < p, \\ \epsilon_j := x_j^p - a_j x_j - x_{j+1} & 1 \leq j \leq n-1, \\ \epsilon_n := x_n^p - a_n x_n. \end{cases}$$

Since $\deg g^{(j)} \leq \deg f$, all the polynomials s_{j,e_j} have degree at most $B-1$. By Lemma 8, the polynomials $\phi\left(\hat{g}^{(j)}(x_j)\right)$ can be computed at degree D. By Lemma 1, all the polynomials s_{j,e_j} can also be computed at that degree.

Since $\hat{g}^{(j-1)}(x_{j-1}) = \gcd\left(f^{(j-1)}(x_{j-1}), \hat{g}^{(j)}(x_{j-1}^p - a_{j-1}x_{j-1})\right)$, there exist polynomials $h^{(j)}$ of degrees at most $p^{j-1} \deg \hat{g}^{(j)} - \deg \hat{g}^{(1)}$ such that

$$\hat{g}^{(j+1)}(L_j(x_1)) = h^{(j)}(x_1)\hat{g}^{(1)}(x_1). \tag{9}$$

where the polynomials L_j are as defined in Section 2.3.

Let us now consider an arbitrary polynomial $u \in I$. We have

$$u(x_1, \ldots, x_n) = v(x_1)\hat{g}^{(1)}(x_1) + \sum_{j=1}^{n-1} q_j(x_1, \ldots, x_n)(x_j^p - a_j x_j - x_{j+1}) + q_n(x_1, \ldots, x_n)(x_n^p - a_n x_n)$$

for some monovariate polynomial v and some polynomials q_j. By successive divisions of v by $h^{(n)}$, $h^{(n-1)}$, etc, we can write

$$v(x_1) = v^{(n+1)}(x_1)h^{(n)}(x_1) + v^{(n)}(x_1)h^{(n-1)}(x_1) + \ldots + v^{(2)}(x_1)h^{(1)}(x_1) + v^{(1)}(x_1)$$

with $\deg v^{(j)} < \deg h^{(j)} \leq p^{j-1} \deg \hat{g}^{(j)} - \deg \hat{g}^{(1)}$. Successively reducing u by ϵ_1, ϵ_2, up to ϵ_n amounts to computing $\phi(u)$. By Equation (9) and the bounds on $\deg v^{(j)}$ we have

$$\begin{aligned} \phi(u) &= \sum_j \phi\left(v^{(j)}(x_1)h^{(j)}(x_1)g^{(1)}(x_1)\right) = \sum_j \sum_{e_j} q_{j,e_j}(x_1, \ldots, x_{j-1})\phi\left(x_j^{e_j} g^{(j)}(x_j)\right) \\ &= \sum_j \sum_{e_j} q_{j,e_j}(x_1, \ldots, x_{j-1}) s_{j,e_j} \end{aligned}$$

for some polynomials q_{j,e_j} with degrees respectively bounded by $(p-1)(j-1)$.

Let now $u \in I$ such that $\deg \phi(u) \leq B$. If the leading term of each term $q_{j,e_j}s_{j,e_j}$ is bounded by B, we are done. Otherwise, there must be at least one cancellation between two leading terms with degree larger than B in the sum. Since $\deg v^{(1)}g^{(1)} < p \deg g^{(2)}$, such a cancellation may not involve any polynomial s_{1,e_1} "of the first block". Let us first assume it involves one polynomial s_{2,e_2} in the second block, and let s_{j,e_j} $(j > 1)$ be the other polynomial involved in this cancellation. The polynomials q_{2,e_2} and q_{j,e_j} may be rewritten as

$$q_{2,e_2} = \sum_{0 \leq e_1 \leq p-1} x_1^{e_1} q_{2,e_2,e_1}(x_2), \quad q_{j,e_j} = \sum_{0 \leq e_1 \leq p-1} x_1^{e_1} q_{j,e_j,e_1}(x_2, \ldots, x_{j-1}).$$

The cancellation necessarily involves terms with the same value of e_1. For this value of e_1, we have

$$LM(x_1^{e_1} q_{2,e_2,e_1}(x_2)) = LM(x_1^{e_1} q_{j,e_j,e_1}(x_2, \ldots, x_{j-1}))$$

where the two polynomials both have degrees larger than B but an appropriate linear combination of them has degree lower or equal to B.

We argue that the linear combination can be computed without increasing the degree above B (even though each polynomial has some terms with degree larger than B). Indeed, a Gröbner basis algorithm can first compute the corresponding linear combination between q_{2,e_2,e_1} and q_{j,e_j,e_1} without the $x_1^{e_1}$ factors. The bound on $\deg v^{(2)}$ tranlates into a bound on $\deg q_{2,e_2,e_1}$ that ensures that this will be possible at degree B. The degree of this linear combination is strictly lower than the original one (including the $x_1^{e_1}$ factor) by e_1. Once this linear combination has been computed, the Gröbner basis algorithm can then compute the linear combination involved in $\phi(u)$ by multiplying by $x_1^{e_1}$. We have therefore shown that any algebraic combination of the equations s_{j,e_j} with degree bounded by B and involving a cancellation of a leading term in the first or second blocks can be computed at degree B (even if the corresponding polynomials in the above decomposition of $\phi(u)$ involve some terms with larger degrees). We proceed similarly with the other blocks to obtain the result.

We deduce the following result:

Proposition 3. *Let f be a monovariate polynomial over $\mathbb{F}_{p^n}$. The degree of regularity of any polynomial system arising from its Weil descent is at most $\lceil (p-1)(\log_p \deg f + 1) \rceil + 1$. The same bound also holds if the system is "hidden" by two bijective linear transformations of variables and equations (as in HFE).*

4 Conclusion and Open Problems

In this paper, we provided a rigourous upper bound on the degree of regularity of polynomial systems arising from a Weil descent of a *monovariate* polynomial f over the finite field $\mathbb{F}_{p^n}$. This proves a conjecture of Petit-Quisquater [21] in the monovariate case, and implies that Gröbner basis algorithms have a quasi-polynomial complexity on this type of systems. We also definitely established similar complexity results on HFE cryptosystems. Although these results have been suspected for more then ten years [12], they have only been "proved" under plausible conjectures or heuristic assumptions so far [15,8,7,1].

In contrast, the proof we presented here holds independently of any heuristic assumption or unproven conjecture. Our approach is radically different from previous ones. The main steps of our proof are heavily connected to (a variant of) SRA, a recently introduced algorithm for finding roots of polynomial equations over an extension field [20]. Not surprisingly, the proof greatly relies on the way these particular systems are generated, from a single polynomial over $\mathbb{F}_{p^n}$ to a polynomial system over $\mathbb{F}_p$. As an additional advantage, the connection to SRA makes our proof completely constructive in contexts were the original polynomial is known. We believe that our variant of SRA is of independent interest.

This paper leaves several more general problems open. Most importantly, an extension of our proof to polynomial systems arising from a Weil descent on a *multivariate* polynomial over $\mathbb{F}_{p^n}$ would be an extremely interesting result. In particular, it would prove that the elliptic curve discrete logarithm over small characteristic fields can be solved in subexponential time, as conjectured in [21]. Another interesting open problem is to consider how the particular structure of some polynomials over $\mathbb{F}_{p^n}$ (besides their degrees) may potentially affect the degree of regularity of their

Weil descent. In particular in the case of HFE polynomials, currently known bounds on the first fall degree [16] as well as experimental results [12] suggest that our upper bound may overestimate the degree of regularity by a factor roughly 2. It will be very interesting to extend our approach to derive better unconditional bounds in this context.

Acknowledgements The author would like to Tim Hodges for carefully reviewing this paper, for pointing out to us an error in a previous version, and for a hand in the proof of Lemma 6.

References

1. Luk 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.
2. Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, Universitt Innsbruck, 1965.
3. Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Springer, 2000.
4. Nicolas Courtois and Jacques Patarin. About the XL algorithm over GF(2). In Marc Joye, editor, *CT-RSA*, volume 2612 of *Lecture Notes in Computer Science*, pages 141–157. Springer, 2003.
5. David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Springer Verlag, Berlin, Heidelberg, New York, 1 edition, 1992.
6. Jintai Ding, Jason E. Gower, and Dieter Schmidt. *Multivariate Public Key Cryptosystems*, volume 25 of *Advances in Information Security*. Springer, 2006.
7. Jintai Ding and Timothy J. Hodges. Inverting HFE systems is quasi-polynomial for all fields. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 724–742. Springer, 2011.
8. Vivien Dubois and Nicolas Gama. The degree of regularity of HFE systems. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 557–576. Springer, 2010.
9. 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.
10. 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.
11. Jean-Charles Faugère, Patrizia M. Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional gröbner bases by change of ordering. *J. Symb. Comput.*, 16(4):329–344, 1993.
12. 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.
13. Jean-Charles Faugère, Ludovic Perret, Christophe Petit, and Guénaël Renault. New subexponential algorithms for factoring in $SL(2, 2^n)$. Cryptology ePrint Archive, Report 2011/598, 2011. `http://eprint.iacr.org/`.
14. Jean-Charles Faugère, Ludovic Perret, Christophe Petit, and Guénaël Renault. Improving the complexity of index calculus algorithms in elliptic curves over binary fields. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 27–44. Springer, 2012.
15. 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.
16. Timothy Hodges, Christophe Petit, and Jacob Schlather. First fall degree and weil descent. Submitted to Finite Fields and their Applications, 2012.
17. Yun-Ju Huang, Christophe Petit, Naoyuki Shinohara, and Tsuyoshi Takagi. Improvement of faugère et al.'s method to solve ecdlp. In Kazuo Sakiyama and Masayuki Terada, editors, *IWSEC*, volume 8231 of *Lecture Notes in Computer Science*, pages 115–132. Springer, 2013.
18. 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.
19. Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In Ueli M. Maurer, editor, *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 1996.
20. Christophe Petit. Finding roots in $\mathbb{F}_{p^n}$ with the successive resultants algorithm. `http://perso.uclouvain.be/christophe.petit/files/SRA.pdf`, 2013.

21. Christophe Petit and Jean-Jacques Quisquater. On polynomial systems arising from a Weil descent. In Xiaoyun Wang and Kazue Sako, editors, *Asiacrypt*, volume 7658 of *Lecture Notes in Computer Science*, pages 451–466. Springer, 2012.
22. Michael Shantz and Edlyn Teske. Solving the elliptic curve discrete logarithm problem using semaev polynomials, weil descent and gröbner basis methods - an experimental study. *IACR Cryptology ePrint Archive*, 2013:596, 2013.