High Order Derivatives and Decomposition of Multivariate Polynomials

Jean-Charles Faugère
Jean-Charles.Faugere@inria.fr

Ludovic Perret
ludovic.perret@lip6.fr

SALSA Project INRIA, Centre Paris-Rocquencourt
UPMC, Univ Paris 06, LIP6
CNRS, UMR 7606, LIP6
UFR Ingénierie 919, LIP6 Passy Kennedy
Boite Courrier 169, 4, Place Jussieu 75252 Paris Cedex 05

ABSTRACT

In this paper, we present an improved method for decomposing multivariate polynomials. This problem, also known as the *Functional Decomposition Problem* (FDP) [17, 9, 27], is classical in computer algebra (e.g. [17, 18, 19, 23, 24, 7, 25]). Here, we propose to use high order partial derivatives to improve the algorithm described in [14]. Our new approach is more simple, and in some sense more natural. From a practical point of view, this new approach will lead to more efficient algorithms. The complexity of our algorithms will depend of the degree of the input polynomials, and the ratio n/u between the number of variables/polynomials.

Categories and Subject Descriptors: I.1.2 Symbolic and Algebraic Manipulation, Algorithms Algebraic Algorithms. **D.4.6 Security and Protection**, Cryptographic controls, Authentication.

General Terms: Algorithms, Security

Keywords: Cryptology, Public Key Cryptography, Gröbner Basis, Algebraic Cryptanalysis, F_5 Algorithm, Multivariate Cryptology.

1. INTRODUCTION

The main concern of this paper is the so-called [17, 9, 27], multivariate *(FDP) Functional Decomposition Problem* (in this paper we do not consider rational decomposition). That is, given a set of u polynomials

$$h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$$

($\mathbb{K}$ denoting an arbitrary field) the goal is to recover – if any – $f = (f_1, \ldots, f_u) \in \mathbb{K}[x_1, \ldots, x_m]^u$ and $g = (g_1, \ldots, g_m) \in \mathbb{K}[x_1, \ldots, x_n]^n$ whose composition equals to h, i.e.:

$$h = \big(h_1, \ldots, h_u\big) = \big(f_1(g_1, \ldots, g_m), \ldots, f_u(g_1, \ldots, g_m)\big).$$

FDP is a classical problem in computer algebra, e.g. [20, 23, 24, 17, 7, 25]. In the univariate case, the decomposition is a standard functionality proposed in several computer algebra systems (for instance, `compoly` of MAPLE[1]). Besides its theoretical interest, the decomposition problem has applications in computer aided geometry [7] and in cryptography [13, 26, 27] for instance. So far, the cryptographic interest of FDP was mainly related to the signature schemes called 2R and 2R$^-$[15, 16]. A public key of such a signature scheme is given by a set by a set of u multivariate polynomials of degree four in n variables. This set of polynomials is obtained from the composition of two secrets systems of quadratic equations. The security of such schemes is then directly related to FDP (when $u \leq n$). From a practical point of view, the algorithm proposed in [13] allowed to completely break the 2R$^-$[15, 16] scheme. Since the cryptographic application was our main motivation we have only considered the restricted case $m = n$. Clearly the case $m \neq n$ deserves also investigations.

1.1 Previous Works

All in all, one can consider that univariate decomposition is a problem which is well mastered from a practical and theoretical point of view (e.g. [1, 2, 20, 23, 24]). However, in the multivariate case, the situation is different. In [17], Von zur Gathen, Gutierrez and Rubio have studied several restrictions of FDP, the *uni-multivariate*, *multi-univariate* and *single-variable* decompositions; which are extensions of the univariate decomposition. From a theoretical point of view, they proved several results on the uniqueness/finiteness of such decompositions. They also proposed an efficient method for decomposing multi-univariate polynomials.

Surprisingly enough, the first technique addressing the multivariate decomposition has been proposed in a very restricted case by Ye, Dai and Lam [26] at CRYPTO'99, a conference in cryptography. More preciely, these authors proposed an efficient algorithm for decomposing a set of n polynomials of degree four into two sets of n quadratic polynomials. Their algorithm essentially used linear algebra techniques. This technique was limited to the special case $u = n$, and for polynomials of degree four.

In [13], the authors of this paper have extended the algorithm presented in [26, 27] for decomposing instances of FDP for which the number of polynomials (u) is smaller or equal than the number of variables (n). To do so, we have used Gröbner bases techniques [5, 6, 3, 4]. However, this algo-

ISSAC'09, July 28–31, 2009, Seoul, Republic of Korea.

[1] `http://www.maplesoft.com/`

rithm remains limited to decompose polynomials of degree four (composition of quadratic polynomials). In [14], we presented an extension of [13] allowing to decompose polynomials of the same arbitrary degree. This last algorithm can be viewed as a recursive version of [13]. In this paper, we will present a new technique for solving FDP. We believe that our new approach is more natural, and leads to more simple algorithms. Besides this "cosmetic" argument, we will see that our new algorithms are more efficient in practice. By the way, with our approach we can solve a bigger class of instances of FDP. Similarly to [13, 14], the complexity of our algorithms will depend of the degrees of the input polynomials, and the ratio n/u.

1.2 Organization of the Paper

The paper is organized as follows. We begin in Section 2 by fixing some notation, introducing the *Functional Decomposition Problem* (FDP) and the class of instances that we will consider. In particular, we will restrict our attention to homogeneous instances of FDP. We call this problem FDP_{H}. In Section 3, we present a new algorithm for FDP_{H}. Briefly, the idea is as follows. Let $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$ be the (homogeneous) polynomials (all of the same degree) obtained from the composition of $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big) \in \mathbb{K}[x_1, \ldots, x_n]^u \times \mathbb{K}[x_1, \ldots, x_n]^n$, i.e.

$$(h_1, \ldots, h_u) = \big(f_1(g_1, \ldots, g_n), \ldots, f_u(g_1, \ldots, g_n)\big).$$

Usually the problem is divided into two steps. First, we have to compute candidates for $g_1, \ldots, g_n$. Then, we can recover $f_1, \ldots, f_u$ from this knowledge. Note that determining $f_1, \ldots, f_u$ knowing $h_1, \ldots, h_u$ and $g_1, \ldots, g_n$ is a subfield membership problem [17, 22]. This is a difficult problem in general. However, in our context, the degree of the polynomials is bounded. Therefore, linear algebra techniques can be used to recover the unknown coefficients of $f_1, \ldots, f_u$.

The harder step is usually to recover candidates for $g_1, \ldots, g_n$. The main contribution of this paper is a new technique for solving this step. More precisely, we will describe an algorithm for recovering the vector space $\mathcal{L}(g) = \text{Span}_{\mathbb{K}}(g_1, \ldots, g_n)$ generated by $g_1, \ldots, g_n$. This vector space will be computed from the reduced DRL Gröbner basis (i.e. a (reduced) Gröbner basis with respect to the degree reverse lexicographical ordering (DRL)) of a suitable ideal. In [14], the authors constructed this set by considering a sequence of quotient ideals constructed from the ideal generated by the partial derivatives of $h_1, \ldots, h_u$, i.e.:

$$\partial\mathcal{I}_h = \left\langle \frac{\partial h_i}{\partial x_j} \mid 1 \leq i \leq u \text{ and } 1 \leq j \leq n \right\rangle.$$

If $h = f \circ g$, we can extract a power of the ideal $\mathcal{I}_g = \langle g_1, \ldots, g_n \rangle$ from $\partial\mathcal{I}_h$. It has been then noticed that recovering $\mathcal{I}_g$ from some power $\mathcal{I}_g^p (p > 1)$ is again a decomposition problem.

When $d = \deg(f) = \max_i(\deg(f_i)) > 2$, we will see in Section 3 that we can speed up the computation by considering the ideal generated by higher order partial derivatives:

$$\partial^k\mathcal{I}_h = \left\langle \frac{\partial^k h_i}{\partial x_{j_1} \cdots \partial x_{j_k}} \mid 1 \leq i \leq u, \text{ and } 1 \leq j_1, \ldots, j_k \leq n \right\rangle,$$

with $k > 0$. This algorithm works in the generic case, as soon as n is sufficiently big (see Theorem 3.4). The algorithm proposed in Section 3 heavily relies on the fact that the input polynomials are all of the same degree. In Section 4, we show that the we can relax this condition and present an algorithm solving a slightly more general version of the problem addressed in Section 3.

2. FUNCTIONAL DECOMPOSITION

We shall say that $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ is a *decomposition* of $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$ if:

$$h = (h_1, \ldots, h_u) = \big(f_1(g_1, \ldots, g_n), \ldots, f_u(g_1, \ldots, g_n)\big) = f \circ g.$$

Note that such decomposition can not be unique. Indeed, any bijective linear combination $A \in \text{GL}_n(\mathbb{K})$ of the polynomials $g_1, \ldots, g_n$ leads to a decomposition of h since:

$$h(x_1, \ldots, x_n) = \big(f(x_1, \ldots, x_n)A^{-1}\big) \circ \big(g(x_1, \ldots, x_n)A\big).$$

This suggests to introduce the following definition [17].

DEFINITION 1. *We shall say that two decompositions* $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ *and* $\big(\tilde{f} = (\tilde{f}_1, \ldots, \tilde{f}_u), \tilde{g} = (\tilde{g}_1, \ldots, \tilde{g}_n)\big)$ *of* $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$ *are equivalent if* $\exists A \in \text{GL}_n(\mathbb{K})$ *such that* $\tilde{g} = g \cdot A$. *Therefore,* h *is uniquely decomposable if all decompositions* (f, g) *of* h *are equivalent.*

We will now discuss the notion of a non trivial decomposition. Indeed, it is worth to remark $\big(f = h, g = (g_1, \ldots, g_n) = (x_1, \ldots, x_n)\big)$, or $\big(f = (x_1, \ldots, x_u), g = (h_1, \ldots, h_u, 0, \ldots, 0)\big)$ are valid, but trivial, decompositions of h. Another case which should be discarded is obtained when considering a polynomial automorphism [8] $(g_1, \ldots, g_n) \in \mathbb{K}[x_1, \ldots, x_n]^n$, i.e. a set for which there exists $(\tilde{g}_1, \ldots, \tilde{g}_n) \in \mathbb{K}[x_1, \ldots, x_n]^n$ such that:

$$x_i = \tilde{g}_i(g_1, \ldots, g_n), \text{for all } i \leq 1 \leq i \leq n.$$

Any $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$ has a decomposition $h = f \circ g$, where f is given by:

$$f_i = h_i(\tilde{g}_1, \ldots, \tilde{g}_n), \text{for all } i \leq 1 \leq i \leq u.$$

From this short discussion, one can understand that it is not obvious to define a proper notion of non trivial decomposition. In [17, 18], the following notion has been proposed.

DEFINITION 2. *Let* $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big) \in \mathbb{K}[x_1, \ldots, x_n]^u \times \mathbb{K}[x_1, \ldots, x_n]^n$ *be a decomposition of* $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$. *This decomposition is non trivial if* $\mathbb{K}[h_1, \ldots, h_u] \subset \mathbb{K}[g_1, \ldots, g_n] \subset \mathbb{K}[x_1, \ldots, x_n]$.

The FDP problem is then equivalent to find a proper intermediate $\mathbb{K}$-algebra between $\mathbb{K}[h_1, \ldots, h_u]$ and $\mathbb{K}[x_1, \ldots, x_n]$. Although this definition is mathematically elegant, we have not been able to use it to study the complexity of our algorithms. For this reason, we have introduced a new notion. We recall that a polynomial is *generic* if its coefficients are considered as variables.

DEFINITION 3. *We shall say that* $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ *is a generic decomposition of* h *if the polynomials of* f *and* g *are generics.*

For a generic decomposition, we conjecture that trivial cases occur only when $d_f = 1$ or $d_g = 1$. To prove this fact it appears to us that a crucial element is missing; which is a multivariate equivalent of Ritt's theorem [21, 11].

In this paper, we will consider a restricted FDP. First, we will assume that the input polynomials are homogeneous of the

same (total) degree. Secondly, we will suppose that the degrees of the output polynomials (f, g) are part of the input. To summarize, let d_f and d_g be integers strictly greater than one.

FDP$_{\mathrm{H}}(d_f, d_g)$
Input: $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$ homogeneous polynomials all of the same degree.
Find: – if any – homogeneous polynomials $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big) \in \mathbb{K}[x_1, \ldots, x_n]^u \times \mathbb{K}[x_1, \ldots, x_n]^n$ of degree d_f and d_g respectively such that $h = f \circ g$.

DEFINITION 4. *We shall say that (f, g) is a (d_f, d_g)-decomposition of h if (f, g) is a decomposition of h, and $\deg(f) = d_f$, $\deg(g) = d_g$.*

From now on, we will always suppose that all the polynomials are homogeneous. Note anyway that our algorithm can be easily extended to the affine case (i.e. non homogeneous case) in the generic situation. Let $p \in \mathbb{K}[x_1, \ldots, x_n]$ and x_0 be a new variable. We denote by $p^*(x_0, x_1, \ldots, x_n) = x_0^{\deg(p)} p(x_1/x_0, \ldots, x_n/x_0)$, the homogenization of p.

LEMMA 2.1. *[14] Let $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big) \in \mathbb{K}[x_1, \ldots, x_n]^u \times \mathbb{K}[x_1, \ldots, x_n]^n$ be such that all the (affine) polynomials of f (resp. g) are of the same degree d_f (resp. d_g). If $\deg(f \circ g) = d_f \cdot d_g$, then :*

$$(f \circ g)^* = f^* \circ g^*,$$

with $f^* = (x_0^{d_f}, f_1^*, \ldots, f_u^*)$ and $g^* = (x_0^{d_g}, g_1^*, \ldots, g_n^*)$.

The condition $\deg(f \circ g) = d_f \cdot d_g$ is always valid for a generic decomposition. Therefore, if (f^*, g^*) is a decomposition of h^*, then a decomposition (f, g) of h is obtained by dehomogenization of f^* and g^*, i.e. by computing $f^*(1, x_1, \ldots, x_n)$ and $g^*(1, x_1, \ldots, x_n)$.

3. ALGORITHM FOR SOLVING FDP$_{\mathrm{H}}$

Let d_f, d_g be integers strictly greater than one. In this part, we will present a new algorithm for solving FDP$_{\mathrm{H}}(d_f, d_g)$. For a *generic* decomposition, we can prove that the algorithm is correct if the decomposition is unique (in the sense of Definition 1) and as soon as n is sufficiently big. For a $(2, 2)$ decomposition ($u = n$), the algorithm works if $n > 4$ (see Theorem 3.4). Our approach is a generalization of the algorithm presented in [13], and an improvement of [14].

3.1 Preliminary Remark

Let (f, g) be a decomposition of h. As already pointed out in the introduction, we can divide the problem in two distinct parts. First, we try to recover the vector space $\mathcal{L}(g) = \mathrm{Span}_{\mathbb{K}}(g_1, \ldots, g_n)$ generated by $g = (g_1, \ldots, g_n)$. Secondly, we find – if any – a decomposition of h by solving a linear system generated from a basis $g = (g_1, \ldots, g_n)$ of $\mathcal{L}(g)$. The "symbolic" equalities :

$$h_i = f_i(g_1, \ldots, g_n), \text{ for all } i, 1 \leq i \leq u, \qquad (1)$$

permit, by comparing the coefficients in the right-most and left-most parts of these equalities, to obtain a linear system of $\mathcal{O}\left(u \cdot \binom{n+d_f-1}{d_f}\right)$ equations in the $u \cdot \binom{n+d_f-1}{d_f}$ unknown coefficients of the $f_1, \ldots, f_u$. Thus, any (non-zero) solution of this linear system will provide a valid decomposition. On the other hand, if this system has no solution, then we can conclude that there exists no valid decomposition. It remains to determine the vector space $\mathcal{L}(g)$.

3.2 Using High Order Derivatives

To explain the basic idea, we first consider FDP$_{\mathrm{H}}(3, 2)$. Let $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ be a $(3, 2)$ decomposition of $h = (h_1, \ldots, h_u)$. By the very definition, it holds that for all $i, 1 \leq i \leq u$:

$$h_i = f_i(g_1, \ldots, g_n) = \sum_{1 \leq k,\ell,m \leq n} f_{k,\ell,m}^{(i)} g_k g_\ell g_m,$$

with $f_i = \sum_{1 \leq k,\ell,m \leq n} f_{k,\ell,m}^{(i)} x_k x_\ell x_m$. We have then:

$$\frac{\partial h_i}{\partial x_j} = \sum_{1 \leq k,\ell,m \leq n} f_{k,\ell,m}^{(i)} \left(g_\ell g_m \frac{\partial g_k}{\partial x_j} + g_k g_m \frac{\partial g_\ell}{\partial x_j} + g_k g_\ell \frac{\partial g_m}{\partial x_j} \right).$$

By considering second order partial derivatives, we get that $\frac{\partial^2 h_i}{\partial x_j \partial x_r} =$

$$\sum_{1 \leq k,\ell,m \leq n} f_{k,\ell,m}^{(i)} \Big(\frac{\partial g_k}{\partial x_j}\frac{\partial g_\ell}{\partial x_r} g_m + \frac{\partial g_\ell}{\partial x_j}\frac{\partial g_m}{\partial x_r} g_k + \frac{\partial g_k}{\partial x_j}\frac{\partial g_m}{\partial x_r} g_\ell + \frac{\partial g_k}{\partial x_r}\frac{\partial g_\ell}{\partial x_j} g_m + \frac{\partial g_\ell}{\partial x_r}\frac{\partial g_m}{\partial x_j} g_k + \frac{\partial g_k}{\partial x_r}\frac{\partial g_m}{\partial x_j} g_\ell + \frac{\partial^2 g_k}{\partial x_j \partial x_r} g_\ell g_m + \frac{\partial^2 g_\ell}{\partial x_j \partial x_r} g_m g_k + \frac{\partial^2 g_m}{\partial x_j \partial x_r} g_k g_\ell \Big).$$

From now on, we will denote by $M(\delta)$ the set of monomials of degree $\delta \geq 0$ in $x_1, \ldots, x_n$. We define the ideal:

$$\partial^2 \mathcal{I}_h = \left\langle \frac{\partial^2 h_i}{\partial x_j \partial x_r} \mid 1 \leq i \leq u, \text{ and } 1 \leq j, r \leq n \right\rangle.$$

We can remark that $\partial^2 \mathcal{I}_h \subseteq \langle x_k \cdot x_\ell \cdot g_m \rangle_{1 \leq k,\ell,m \leq n}$. In general, the ideal $\partial^2 \mathcal{I}_h$ provides enough information to recover $\mathcal{L}(g)$.
Let:

$$\widetilde{V} = \left\{ \frac{\partial^2 h_i}{\partial x_j \partial x_r} \mid 1 \leq i \leq u, \text{ and } 1 \leq j, r \leq n \right\}.$$

As explained below, each second order partial derivative $\frac{\partial^2 h_i}{\partial x_j \partial x_r} \in \widetilde{V}$ is a linear combination of elements:

$$\{x_k \cdot x_\ell \cdot g_m\}_{1 \leq k,\ell,m \leq n}.$$

We can construct a matrix as follows:

$$A_{\widetilde{V}} = \begin{array}{c} \\ \vdots \\ \vdots \\ \frac{\partial^2 h_i}{\partial x_j \partial x_r} \\ \vdots \\ \vdots \end{array} \begin{array}{c} \cdots \quad \cdots \quad x_k \cdot x_m \cdot g_\ell \quad \cdots \quad \cdots \\ \begin{pmatrix} & \cdots & \\ & \cdots & \\ & \cdots & \\ & \cdots & \\ & \cdots & \end{pmatrix} \end{array}$$

where each coefficient corresponds to the coefficient of a $x_k \cdot x_\ell \cdot g_m$ in $\frac{\partial^2 h_i}{\partial x_j \partial x_r}$. We remark that if $\mathrm{Rank}(A_{\widetilde{V}}) = |\mathrm{Columns}(A_{\widetilde{V}})| = n^2|M(2)|$, then[2] each $x_k x_\ell g_m$ can be expressed as a linear combination of $\frac{\partial^2 h_i}{\partial x_j \partial x_r}$. In particular:

$$x_n^2 g_i \in \mathrm{Vect}_{\mathbb{K}}(\widetilde{V}) \subset \partial^2 \mathcal{I}_h, \text{for all } i, 1 \leq i \leq n.$$

We can observe that $\dim_{\mathbb{K}}\big(\mathrm{Vect}_{\mathbb{K}}(\widetilde{V})\big)$ is upper-bounded by $u^2|M(2)|$. This equality can occur only if $u = 2$. When $\mathrm{Rank}(A_{\widetilde{V}}) \neq n^2|M(2)|$, the idea is to consider a bigger vector space by *stretching* the matrix.

[2] the notation $|.|$ stands for the cardinality of a set.

More precisely, we multiply each $\frac{\partial^2 h_i}{\partial x_j \partial x_r}$ by a monomial m of degree $\delta \geq 0$. One can see that each polynomial of:

$$\widetilde{V}_\delta = \left\{ m \frac{\partial^2 h_i}{\partial x_j \partial x_r} \mid m \in M(\delta), 1 \leq i \leq u, \text{ and } 1 \leq j, r \leq n \right\},$$

is a linear combination of elements:

$$\{ m' g_k \mid m' \in M(\delta + 2), \text{ and } 1 \leq k \leq n \}.$$

We define a new matrix:

$$A_{\widetilde{V}_\delta} = m \frac{\partial^2 h_i}{\partial x_j \partial x_r} \begin{array}{c} \\ \vdots \\ \\ \\ \\ \vdots \\ \\ \vdots \end{array} \begin{array}{c} \cdots \quad \cdots \quad m' g_k \quad \cdots \quad \cdots \\ \begin{pmatrix} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \end{pmatrix} \end{array}$$

Similarly, $x_n^{\delta+2} g_i \in \mathrm{Vect}_{\mathbb{K}}(\tilde{V}_\delta) \subset \partial^2 \mathcal{I}_h$, for all $i, 1 \leq i \leq n$, if:

$$\mathrm{Rank}(A_{\widetilde{V}_\delta}) = |\mathrm{Columns}(A_{\widetilde{V}_\delta}| = n|M(\delta + 2)|.$$

To generalize this approach to $\mathrm{FDP}_{\mathrm{H}}(d_f, d_g)$, with $d_f \geq 2$, we will use the following trivial result.

LEMMA 3.1. *Let $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ be a (d_f, d_g) decomposition of $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$. We set $\partial^{(d_f-1)} \mathcal{I}_h = \left\langle \frac{\partial^{(d_f-1)} h_i}{\partial x_{j_1} \cdots \partial x_{j_{(d_f-1)}}} \mid 1 \leq i \leq u, \text{ and } 1 \leq j_1, \ldots, j_{(d_f-1)} \leq n \right\rangle$.*

We have that each polynomial of $\partial^{(d_f-1)} \mathcal{I}_h$ is of the form:

$$\sum_{\ell=1}^{n} H_\ell(x_1, \ldots, x_n)\, g_\ell,$$

where each H_ℓ is 0 or a polynomial of degree $(d_g - 1)(d_f - 1)$.

PROOF. Since, for all $i, 1 \leq i \leq u$, f_i is a sum of monomials of degree d_f it follows that $h_i = f_i \circ g$ is a sum of products of the following shape:

$$g_{j_1} \cdots g_{j_{d_f}}$$

Obviously, any partial derivative of order $d_f - 1$ of such an expression can always be written in the following form :

$$\frac{\partial^{(d_f-1)}}{\partial x_{k_1} \cdots \partial x_{k_{(d_f-1)}}} \left(g_{j_1} \cdots g_{j_{d_f}} \right) = \sum_{\ell=1}^{d_f} h_\ell(x_1, \ldots, x_n)\, g_{j_\ell},$$

where h_ℓ is 0 or a polynomial of degree $(d_g - 1)(d_f - 1)$. The remaining claims of the proof follow easily. □

We are now in position to present the main result of this section. According to Lemma 3.1, each polynomial :

$$m \frac{\partial^{(d_f-1)} h_i}{\partial x_{j_1} \cdots \partial x_{j_{(d_f-1)}}}, m \in M(\delta),$$

is a linear combination of elements:

$$\{ m' g_k \mid m' \in M(\delta + d), \text{ and } 1 \leq k \leq n \},$$

with $d = (d_f - 1)(d_g - 1)$. It then makes sense to consider the matrix $A_{\widetilde{V}_\delta}$ defined as follows :

$$m \frac{\partial^{(d_f-1)} h_i}{\partial x_{j_1} \cdots \partial x_{j_{(d_f-1)}}} \begin{array}{c} \\ \vdots \\ \\ \\ \\ \vdots \\ \\ \vdots \end{array} \begin{array}{c} \cdots \quad \cdots \quad m' g_k \quad \cdots \quad \cdots \\ \begin{pmatrix} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \end{pmatrix} \end{array} \tag{2}$$

with $m \in M(\delta)$, and $m' \in M\big(\delta + (d_f - 1)(d_g - 1)\big)$. We can then fully extend the approach.

THEOREM 3.1. *Let $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ be a (d_f, d_g) decomposition of $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$. Let $\partial^{(d_f-1)} \mathcal{I}_h$ be defined as in Lemma 3.1, $A_{\widetilde{V}_\delta}$ be the matrix (2), and $d = (d_f - 1)(d_g - 1)$. If $\mathrm{Rank}(A_{\widetilde{V}_\delta}) = n|M(\delta + d)|$, for some $\delta \geq 0$, then :*

$$x_n^{\delta+d} g_i \in \partial^{(d_f-1)} \mathcal{I}_h, \text{ for all } i, 1 \leq i \leq n. \tag{3}$$

PROOF. As already explained, $\mathrm{Rank}(A_{\widetilde{V}_\delta}) = n|M(\delta + d)|$, for some $\delta \geq 0$, implies that:

$$x_n^{\delta+d} g_i \in \mathrm{Vect}_{\mathbb{K}}(\widetilde{V}_\delta) \subset \partial^{(d_f-1)} \mathcal{I}_h, \text{ for all } i, 1 \leq i \leq n.$$

This concludes the proof of the theorem. □

An important parameter in Theorem 3.1 is the value of δ. For a generic decomposition, we can determine a lower bound.

3.2.1 Analysis of δ

We consider a generic (d_f, d_g) decomposition $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ of $h = (h_1, \ldots, h_u)$. In this case, the fact that there exists $\delta \geq 0$ s.t. $\mathrm{Rank}(A_{\widetilde{V}_\delta}) = n|M(\delta + d)|$ is equivalent to:

$$|\widetilde{V}_\delta| \geq |\{ m' g_k \mid m' \in M(\delta + d), \text{ and } 1 \leq k \leq n \}|, \text{for a } \delta \geq 0.$$

We have that:

$$|\widetilde{V}_\delta| = u \cdot |M(\delta)||M(d_f - 1)|.$$

There exists $\delta \geq 0$ s.t. $\mathrm{Rank}(A_{\widetilde{V}_\delta}) = n|M(\delta + d)|$ if:

$$\frac{u}{n} \cdot |M(d_f - 1)| \frac{|M(\delta)|}{|M(\delta + d)|} \geq 1, \text{for some } \delta \geq 0.$$

For $d_f = d_g = 2$, we obtain for instance that $\delta \geq \frac{n-1}{u-1}$.

3.2.2 Computing $\mathcal{L}(g)$

We will now explain how we can recover the set $\mathcal{L}(g)$ defined in Theorem 3.1. To do so, we remark that (3) implies:

$$\mathcal{L}(g) \subset \langle g_1, \ldots, g_n \rangle \subseteq \partial^{(d_f-1)} \mathcal{I}_h : \langle x_n^{\delta+d} \rangle.$$

The ideal $\partial^{(d_f-1)} \mathcal{I}_h$ being homogenous, its quotient (or column) ideal is also an homogeneous ideals. As soon as (3) is valid, it holds that$\min\big(\deg(p) \mid p \in \partial^{(d_f-1)} \mathcal{I}_h : \langle x_n^{\delta+d} \rangle\big) = d_g$. Let G be a reduced DRL Gröbner basis of $\partial^{(d_f-1)} \mathcal{I}_h : \langle x_n^{\delta+d} \rangle$ and $B_g = \big\{ g \in G \mid \deg(g) = d_g \big\}$. According to the minimality – w.r.t. the degree – of a DRL Gröbner basis :

$$\mathrm{Span}_{\mathbb{K}}\big(B_g\big) = \mathrm{Span}_{\mathbb{K}}\big(g \in \partial^{(d_f-1)} \mathcal{I}_h : \langle x_n^{\delta+d} \rangle \mid \deg(g) = d_g\big).$$

In particular, we have that $\mathcal{L}(g) \subseteq \mathrm{Span}_{\mathbb{K}}\big(B_g\big)$. The equality holds if $\dim_{\mathbb{K}}\big(\mathrm{Span}_{\mathbb{K}}(B_g)\big) = n$. This conditions implies in particular that the decomposition is unique.

Note that a DRL Gröbner basis G of $\partial^{(d_f-1)} \mathcal{I}_h : \langle x_n^{\delta+d} \rangle$ can be computed using standard elimination techniques [8]. In our situation, an alternative method can be used. This is due to the particular

role of x_n in a DRL order. Namely, if x_n^p (with $p > 0$) divides the leading monomial of a polynomial, then it also divides the whole polynomial. Thus, we can restrict our attention to polynomials of a DRL Gröbner basis G' of $\partial^{(d_f-1)}\mathcal{I}_h$ whose leading monomials contains $x_n^{\delta+d}$. More precisely, $\{g \in G \mid \deg(g) = d_g\} =$

$$\left\{ \frac{g'}{x_n^{\delta+d}} \mid g' \in G', \text{ and } x_n^{\delta+d} \mid \mathrm{LM}(g', \prec_{DRL}) \right\}.$$

It is then sufficient to compute a reduced DRL Gröbner basis up to the degree $(\delta + d + d_g)$ of the homogeneous ideal $\partial^{(d_f-1)}\mathcal{I}_h$ to compute a DRL Gröbner basis up to the degree d_g of $\partial^{(d_f-1)}\mathcal{I}_h : \langle x_n^{\delta+d} \rangle$. In the rest of this paper, we shall call d-Gröbner basis a Gröbner basis up to the degree d.

3.3 Description of the Algorithm

We are now in position to describe our algorithm.

MultiComPoly

Input : integers $d_f, d_g > 1$; homogenous polynomials $h_1, \ldots, h_u$ of the same degree d_h

Output : `Fail`, or homogeneous polynomials $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ of degree d_f and d_g resp. such that $h = f \circ g$.

$d \leftarrow (d_f - 1)(d_g - 1)$
$\delta \leftarrow \min_{\delta' \geq 0} \left\{ \frac{u}{n} \cdot |M(d_f - 1)| \frac{|M(\delta')|}{|M(\delta'+d)|} \geq 1 \right\}$.
$G \leftarrow$ a d_g-DRL Gröbner basis of $\partial^{(d_f-1)}\mathcal{I}_h : \langle x_n^{\delta+d} \rangle$
If $\dim_{\mathbb{K}}\big(\mathrm{Span}_{\mathbb{K}}(G)\big) \neq n$ **then** Return `Fail`
Let $g = (g_1, \ldots, g_n)$ be a basis of $\mathrm{Span}_{\mathbb{K}}(G)$
Compute Sys the set of solutions of the linear system generated, as explained in (1), from g
If $|Sys| = 0$ **then**
  Return `Fail` `// no non trivial decomposition`
Else Pick a random element $f = (f_1, \ldots, f_u)$ of Sys
  Return $\big(g = (g_1, \ldots, g_n), f = (f_1, \ldots, f_u)\big)$

REMARK 3.1. *In this form, we will see that the algorithm permits to solve generic instances of FDP$_{\mathrm{H}}$. For non-generic instances, it can be possible to decompose. But, you have to increase the value of δ. Namely, we have to choose:*

$$\delta > \min_{\delta' \geq 0} \left\{ \frac{u}{n} \cdot |M(d_f - 1)| \frac{|M(\delta')|}{|M(\delta' + d)|} \geq 1 \right\}.$$

To handle these cases, we can modify the algorithm by adding a loop on the value of δ. It has to be noted that the algorithm will return `Fail` *as soon as the decomposition is not unique. In this case, we have necessarily* $\dim_{\mathbb{K}}\big(\mathrm{Span}_{\mathbb{K}}(G)\big) > n$.

We now study the complexity of our algorithm.

THEOREM 3.2. *Let $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ be a (d_f, d_g) decomposition of $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$. We set The complexity of* ***MultiComPoly*** *is $\mathcal{O}\big(n^{3(\delta+d+d_g)}\big)$.*

PROOF. We assume that the complexity of **MultiComPoly** is dominated by the cost of computing a reduced DRL Gröbner basis G of $\partial^{(d_f-1)}\mathcal{I}_h : \langle x_n^{\delta+d} \rangle$. However, as explained previously, it is sufficient to compute a reduced $(\delta + d + d_g)$-DRL Gröbner basis of the homogeneous ideal $\partial^{(d_f-1)}\mathcal{I}_h : \langle x_n^{\delta+d} \rangle$. This can be done with the F$_5$ algorithm [12] in $\mathcal{O}\big(n^{3(\delta+d+d_g)}\big)$. □

We will now study the behavior of **MultiComPoly** in the generic situation.

THEOREM 3.3. *Let $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ be a (d_f, d_g) generic decomposition of $h = (h_1, \ldots, h_u)$. If* ***Multi-ComPoly****(d_f, d_g) with the FGb it returns a (d_f, d_g) decomposition (f', g') which is equivalent to (f, g).*

PROOF. According to the Theorem 3.1 and Section 3.2.1, the set G computed in **MultiComPoly** verifies :

$$\mathcal{L}(g) \subseteq \mathrm{Span}_{\mathbb{K}}(G).$$

By hypothesis, we have that $\dim_{\mathbb{K}}\big(\mathrm{Span}_{\mathbb{K}}(G)\big) = n$ (otherwise, the algorithm would have returned `Fail`). Therefore, $\mathcal{L}(g) = \mathrm{Span}_{\mathbb{K}}(G)$ and the polynomials g' are a basis of $\mathcal{L}(g)$. This implies that $\exists A \in \mathrm{GL}_n(\mathbb{K})$ such that $g' = g \cdot A$. □

For small values of n, we have observed that the algorithm will always return `Fail` (independently of the fact that a decomposition exits or not). To explain this phenomenon, we consider a $(2, 2)$ decomposition (with $u = n$) :

THEOREM 3.4. *Let $h = (h_1, \ldots, h_n)$ be generic polynomials of degree four.* ***MultiComPoly****$(2, 2, h)$ will always return* `Fail` *if:*

$$|M(3)| - 2 \cdot n < n^2.$$

PROOF. We will suppose that the $\frac{\partial h_i}{\partial x_j} \in \partial\mathcal{I}_h$ are generic polynomials. We construct the matrix H representing the polynomials $\frac{\partial h_i}{\partial x_j}$ in a basis of monomials, i.e. where each coefficient of H corresponds to the coefficient of the monomial $m \in M(3)$ in $\frac{\partial h_i}{\partial x_j}$. The matrix H is as follows :

$$\begin{array}{c} \\ \frac{\partial h_i}{\partial x_j} \end{array} \begin{array}{c} \cdots \quad \cdots \quad m \in M(3) \quad \cdots \quad \cdots \\ \left(\begin{array}{ccccc} & & \cdots & & \\ & & \cdots & & \\ & & \cdots & & \\ & & \cdots & & \\ & & \cdots & & \end{array} \right) \end{array}$$

This matrix has $u \cdot n$ rows and $|M(3)|$ columns. We can suppose that the monomials are sorted w.r.t. the DRL. Therefore the last $n(n+1)/2$ columns correspond to monomials which can be divided by x_n. These monomials can be splitted in two sets : $x_n^2 x_i$, with $1 \leq i \leq n$ and $x_n x_i x_j$, $i, j, 1 \leq i \leq j \leq (n-1)$. Let G be the set computed in **MultiComPoly**. In our case the value of δ is 0. The algorithm succeeds if the set $\dim_{\mathbb{K}}\big(\mathrm{Span}_{\mathbb{K}}(G)\big) = n$. The elements of G are obtained from the polynomials of degree 3 in $\mathcal{I}_h$ of the form x_n times a polynomial of degree 2. If $|M(3)| - 2 \cdot n < n^2$, then we have necessarily more that n elements in G, and the algorithm will `Fail`. □

In particular, we get that **MultiComPoly**$(2, 2, h)$ will always return `Fail` if $n < 5$. This is exactly what we have observed in practice. We can generalize this analysis for a (d_f, d_g) decomposition. However, the bound is not tight. This is likely due to the fact that we can no longer suppose that high order derivatives behave as generic polynomials (as assumed in the proof). But, when n is too small, the algorithm will always return `Fail` (independently of the fact that a decomposition exits or not).

3.3.1 Experimental Results

To complete the analysis, we present experimental results obtained with **MultiComPoly** on "*random*" decomposable instances of FDP$_{\mathrm{H}}$. Namely, we have randomly selected homogeneous polynomials $f = (f_1, \ldots, f_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$ of degree d_f with $d_f = 3, 5$. We have randomly chosen homogeneous quadratic polynomials $g = (g_1, \ldots, g_n)$. We then have tried to decompose:

$$h = \big(h_1, \ldots, h_u\big) = \big(f_1(g_1, \ldots, g_n), \ldots, f_u(g_1, \ldots, g_n)\big).$$

We have generated these instances using MAPLE. For the Gröbner bases computations, we have used the F$_5$ version implemented in C with the FGb software[3]. We have considered the polynomial ring

[3] http://fgbrs.lip6.fr/jcf/Software/FGb/index.html

$\mathbb{F}_p[x_1,\ldots,x_n]$ with $p \approx 2^{32}$. In the table, we have quoted $\delta_{\exp}$, which is the smallest $\delta \geq 0$ s. t.:

$$\dim_{\mathbb{K}}\left(\mathrm{Vect}_{\mathbb{K}}\left(p \in \partial^{(d_f-1)}\mathcal{I}_h : x_n^{d+\delta} \mid \deg(p) = d_g\right)\right).$$

For comparison, we have also quoted the theoretical value that you should obtain for a generic decomposition, namely:

$$\delta_{\mathrm{theo}} = \min_{\delta' \geq 0}\left\{\frac{u}{n} \cdot |M(d_f-1)|\frac{|M(\delta')|}{|M(\delta'+d)|} \geq 1\right\}.$$

We have denoted $\mathsf{T}_{\mathrm{New}}$ the time needed for computing a d_g-Gröbner basis of $\partial^{(d_f-1)}\mathcal{I}_h : x_n^{d+\delta_{\exp}}$. We have not included the time needed for generating instances. Finally, $\mathsf{T}_{\mathrm{Old}}$ is the time that we obtained with the algorithm presented in [14] (only the Gröbner bases computations). We have the following results:

u	n	d_f	$\delta_{\exp}$	δ_{theo}	$x_n^{d+\delta_{\exp}}$	$\mathsf{T}_{\mathrm{New}}$	$\mathsf{T}_{\mathrm{Old}}$
6	6	3	0	0	x_n^2	0.01 s.	0.12 s.
7	7	3	0	0	x_n^2	0.02 s.	0.33 s.
8	8	3	0	0	x_n^2	0.07 s.	0.86 s.
9	9	3	0	0	x_n^2	0.16 s.	2.38 s.
10	10	3	0	0	x_n^2	0.33 s.	5.81 s.
11	11	3	0	0	x_n^2	0.75 s.	13.8 s.
12	12	3	0	0	x_n^2	1.75 s.	31.6 s.
13	13	3	0	0	x_n^2	3.02 s.	69.7 s.
14	14	3	0	0	x_n^2	5.7 s.	146.4 s.
15	15	3	0	0	x_n^2	10.7 s.	311.4 s.
16	16	3	0	0	x_n^2	19.2 s.	577.9 s.
17	17	3	0	0	x_n^2	33.6 s.	1110.9 s.
4	7	3	1	1	x_n^3	0.04 s.	0.21 s.
4	8	3	1	1	x_n^3	0.1 s.	0.51 s.
5	9	3	1	1	x_n^3	0.4 s.	1.59 s.
5	10	3	1	1	x_n^3	1.3 s.	1.78 s.
6	12	3	1	1	x_n^3	11.2 s.	22.6 s.
7	14	3	1	1	x_n^3	69 s.	119.5 s.
8	16	3	1	1	x_n^3	341.3 s.	
9	18	3	1	1	x_n^3	1349.3 s.	
10	20	3	1	1	x_n^3	9688.9 s.	
7	7	4	0	0	x_n^3	0.16 s.	
8	8	4	0	0	x_n^3	0.65 s.	
9	9	4	0	0	x_n^3	0.93 s.	
10	10	4	0	0	x_n^3	7.5 s.	

We have observed that $\delta_{\exp}$ is always equal to the δ_{theo} predicted. Note also that we have quoted the minimum values of u and n for which the algorithm can return a decomposition. For instance, in $(3,2)$ decomposition $(u=n)$, the algorithm will always return `Fail` if $n<6$. In addition, we emphasize that our new algorithm is more efficient in practice than the one presented in [14]. The first difference between [14] and our algorithm is that we only compute here one Gröbner basis. In [14], we have to compute d_f-1 Gröbner bases. In [14], we have only considered first order partial derivatives. Thus, we deal with equations of higher degree than in our case (here, we have considered high order partial derivatives). This influences the cost of the Gröbner bases computations. To summarize, our new algorithm performs fewer steps, and the cost of the computation of Gröbner bases is cheaper.

4. AN EXTENSION OF THE ALGORITHM

Until now, we have supposed that all the polynomials $h_1,\ldots,h_u$ are of the same degree. In this part, we present a simple extension of **MultiComPoly** which permits to decompose $h=(h_1,\ldots,h_u)$ of arbitrary degrees when the decomposition $f \circ g$ of h is s.t.:

- all the polynomials of $g=(g_1,\ldots,g_n)$ are homogeneous of the same degree d_g, and
- the polynomials $(f_1,\ldots,f_u)$ are homogeneous of degrees $d_{f_1},\ldots,d_{f_u}$ respectively.

To summarize, let $d_{f_1},\ldots,d_{f_u},d_g$ be integers >1. We will present an algorithm for solving :

FDP$_{\mathrm{H}}([d_{f_1},\ldots,d_{f_u}],d_g)$
Input : $h=(h_1,\ldots,h_u) \in \mathbb{K}[x_1,\ldots,x_n]^u$.
Find : – if any – homogeneous polynomials $\big(f=(f_1,\ldots,f_u),$ $g=(g_1,\ldots,g_n)\big) \in \mathbb{K}[x_1,\ldots,x_n]^u \times \mathbb{K}[x_1,\ldots,x_n]^n$ of degree $[d_{f_1},\ldots,d_{f_u}]$ and d_g respectively such that $h=f\circ g$.

Using obvious notation, we shall say that (f,g) is a $([d_{f_1},\ldots,d_{f_u}],$ $d_g)$-*decomposition* of h if (f,g) is a decomposition of h, and $\deg(f_i)=d_{f_i}$, for all $i, 1\leq i\leq u$ and $\deg(g)=d_g$. For such decompositions, we can extend the algorithm **MultiComPoly**. To do so, we remark that $\mathcal{I}_h=\langle h_1,\ldots h_u\rangle$ is a disjoint union of homogeneous ideals, namely:

$$\mathcal{I}_h = \cup_{i=1}^u \mathcal{I}_h^{(i)},$$

where $\mathcal{I}_h^{(i)}$ is generated by homogeneous polynomials of degree $d_{f_i}d_g$. According to Lemma 3.1, each polynomial

$$\frac{\partial^{(d_{f_i}-1)}h_i}{\partial x_{j_1}\cdots\partial x_{j_{(d_{f_i}-1)}}}, \text{with } \deg(h_i)=d_{f_i}d_g,$$

is of the form:

$$\sum_{\ell=1}^{n} H_\ell(x_1,\ldots,x_n)\, g_\ell,$$

where each H_ℓ is 0 or a polynomial of degree $d_i=(d_g-1)(d_{f_i}-1)$. Let $d=\max_{1\leq i\leq u}\big((d_{f_i}-1)(d_g-1)\big)$. For all $i, 1\leq i\leq u$, each element of the set $\widetilde{V}_\delta^{(i)}=$

$$\left\{m\frac{\partial^{(d_{f_i}-1)}h_i}{\partial x_{j_1}\cdots\partial x_{j_{(d_{f_i}-1)}}} \mid \deg(h_i)=d_{f_i}d_g \text{ and } m\in M(\delta+d-d_i)\right\},$$

is a linear combination of elements:

$$\big\{m'g_k \mid m'\in M(\delta+d), \text{ and } 1\leq k\leq n\big\}.$$

We can consider the matrix $A_{\widetilde{V}_\delta}$ whose rows are indexed by the elements of $\cup_{1\leq i\leq u}\widetilde{V}_\delta^{(i)}$ and columns by the elements of $\big\{m'g_k \mid m'\in M(\delta+d), \text{ and } 1\leq k\leq n\big\}$. Namely, $A_{\widetilde{V}_\delta}$ is a follows:

$$\begin{array}{c} \\ p\in\bigcup_{1\leq i\leq u}\widetilde{V}_\delta^{(i)} \end{array} \begin{array}{c} \begin{array}{ccccc} \cdots & \cdots & m'g_k & \cdots & \cdots \end{array} \\ \begin{pmatrix} & & \cdots & & \\ & & \cdots & & \\ & & \cdots & & \\ & & \cdots & & \\ & & \cdots & & \end{pmatrix} \end{array} \tag{4}$$

with $m'\in M(\delta+d)$. We then have:

THEOREM 4.1. *Let $\big(f=(f_1,\ldots,f_u), g=(g_1,\ldots,g_n)\big)$ be a $([d_{f_1},\ldots,d_{f_u}],d_g)$ decomposition of $h=(h_1,\ldots,h_u)\in\mathbb{K}[x_1,\ldots,x_n]^u$. Let $\mathcal{I}_h=\langle h_1,\ldots h_u\rangle$ and $\mathcal{I}_h=\cup_{i=1}^u\mathcal{I}_h^{(i)}$, where $\mathcal{I}_h^{(i)}$ is the ideal generated by polynomials of degree $d_{f_i}d_g$. We define* $\partial^{(d_{f_i}-1)}\mathcal{I}_h^{(i)}=$

$$\left\langle \frac{\partial^{(d_{f_i}-1)}h}{\partial x_{j_1}\cdots\partial x_{j_{(d_{f_i}-1)}}} \mid h\in\mathcal{I}_h^{(i)}, \text{ and } 1\leq j_1,\ldots,j_{(d_{f_i}-1)}\leq n\right\rangle,$$

and $D(\mathcal{I}_h)=\cup_{i=1}^u\partial^{(d_{f_i}-1)}\mathcal{I}_h^{(i)}$. Let $A_{\widetilde{V}_\delta}$ be the matrix (4)*. We set $d=\max_{1\leq i\leq u}\big((d_{f_i}-1)(d_g-1)\big)$. If $\mathrm{Rank}(A_{\widetilde{V}_\delta})=n|M(\delta+d)|$, for some $\delta\geq 0$, then:*

$$\mathcal{L}(g)\subset D(\mathcal{I}_h):\langle x_n^{\delta+d}\rangle.$$

PROOF. If $\mathrm{Rank}(A_{\widetilde{V}_\delta})=n|M(\delta+d)|$, for some $\delta\geq 0$, then :

$$x_n^{\delta+d}g_i\in\mathrm{Vect}_{\mathbb{K}}(\cup_{1\leq i\leq u}\widetilde{V}_\delta^{(i)})\subset D(\mathcal{I}_h), \text{ for all } i, 1\leq i\leq n.$$

Therefore $\mathcal{L}(g)\subset\langle g_1,\ldots,g_n\rangle\subseteq D(\mathcal{I}_h):\langle x_n^{\delta+d}\rangle$. □

As explained in 3.2.2, the set $\mathcal{L}(g)$ can be recovered from a $(\delta + d + d_g)$-Gröbner basis of $D(\mathcal{I}_h)$. As in the previous section, we can obtain a bound for δ by considering the matrix defined in (4).

PROPERTY 4.1. *Let* $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ *be a generic* $([d_{f_1}, \ldots, d_{f_u}], d_g)$ *decomposition of* $h = (h_1, \ldots, h_u)$. *We denote* $d = \max_{1 \leq i \leq u} \big((d_{f_i} - 1)(d_g - 1)\big)$. *Therefore,* $\exists \delta \geq 0$, *such that* $\mathrm{Rank}(A_{\widetilde{V}_\delta}) = n|M(\delta + d)| \iff \exists \delta \geq 0$ *such that :*

$$\frac{\sum_{i=1}^{u} u_i \cdot |M(\delta + d - d_i)||M(d_{f_i} - 1)|}{n|M(\delta + d)|} \geq 1,$$

with u_i *being the number of polynomials* h_i *of degree* $d_{f_i} d_g$.

PROOF. The proof uses the same arguments than the ones used in Section 3.2.1. We have that $|\widetilde{V}_\delta^{(i)}| = u_i \cdot |M(\delta + d - d_i)| n^{d_{f_i} - 1}$. The fact that there exists $\delta \geq 0$ s.t. $\mathrm{Rank}(A_{\widetilde{V}_\delta}) = n|M(\delta + d)|$ is equivalent to the fact that there exists $\delta \geq 0$ s. t. :

$$|\cup_{1 \leq i \leq u} \widetilde{V}_\delta^{(i)}| \geq |\{m' g_k \mid m' \in M(\delta + d), \text{ and } 1 \leq k \leq n\}|.$$

It holds that $|\cup_{1 \leq i \leq u} \widetilde{V}_\delta^{(i)}| = \sum_{i=1}^{u} u_i \cdot |M(\delta + d - d_i)||M(d_{f_i} - 1)|$. So, there exists $\delta \geq 0$ s.t. $\mathrm{Rank}(A_{\widetilde{V}_\delta}) = n|M(\delta + d)| \iff$

$$\frac{\sum_{i=1}^{u} u_i \cdot |M(\delta + d - d_i)||M(d_{f_i} - 1)|}{n|M(\delta + d)|} \geq 1, \text{for some } \delta \geq 0.$$

This concludes the proof. □

The description of the extended algorithm is finally very similar to **MultiComPoly**.

MultiComPoly$_1$
Input :
- integers $([d_{f_1}, \ldots, d_{f_u}], d_g)$ all > 1
- homogenous polynomials $h_1, \ldots, h_u$ of distinct degrees

Output :
- `Fail`, or
- homogeneous polynomials $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ of degrees $[d_{f_1}, \ldots, d_{f_u}]$ and d_g resp. such that $h = f \circ g$.

$G \leftarrow \emptyset$ and $\mathcal{I}_h \leftarrow \langle h_1, \ldots, h_u \rangle$
$D(\mathcal{I}_h) \leftarrow \cup_{i=1}^{u} \partial^{(d_{f_i} - 1)} \mathcal{I}_h^{(i)}$
$d \leftarrow \max_{1 \leq i \leq u} \big((d_{f_i} - 1)(d_g - 1)\big)$
$\delta \leftarrow \min_{\delta' \geq 0} \left\{ \frac{\sum_{i=1}^{u} u_i \cdot |M(\delta' + d - d_i)||M(d_{f_i} - 1)|}{n|M(\delta' + d)|} \geq 1 \right\}$.
$G \leftarrow$ a d_g-DRL Gröbner basis of $D(\mathcal{I}_h) : \langle x_n^{\delta + d} \rangle$
If $\dim_{\mathbb{K}} \big(\mathrm{Span}_{\mathbb{K}}(G)\big) \neq n$ **then** Return `Fail`
Let $g = (g_1, \ldots, g_n)$ be a basis of $\mathrm{Span}_{\mathbb{K}}(G)$
Compute Sys the set of solutions of the linear system generated, as explained in (1), from g
If $|Sys| = 0$ **then**
  Return `Fail` `// no non trivial decomposition`
Else Pick a random element $f = (f_1, \ldots, f_u)$ of Sys
  Return $\big(g = (g_1, \ldots, g_n), f = (f_1, \ldots, f_u)\big)$

The remark 3.1 is also valid for **MultiComPoly**$_1$. For the same reasons explained in the previous section, this algorithm will always return `Fail` is n is too small. For the complexity:

THEOREM 4.2. *Let* $\big(f = (f_1, \ldots, f_u), g = (g_1, \ldots, g_n)\big)$ *be a* $([d_{f_1}, \ldots, d_{f_u}], d_g)$ *decomposition of* $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$. *The complexity of* ***MultiComPoly***$_1$ *is :*

$$\mathcal{O}\big(n^{3(\delta + d + d_g)}\big).$$

PROOF. The complexity of **MultiComPoly**$_1$ is dominated by the cost of computing a reduced DRL Gröbner basis G of $D(\mathcal{I}_h) : \langle x_n^{\delta + d} \rangle$. This basis can be constructed from a reduced $(\delta + d + d_g)$-DRL Gröbner basis $D(\mathcal{I}_h)$. This can be done with F_5 [12] in $\mathcal{O}\big(n^{3(\delta + d + d_g)}\big)$. □

4.1 Experimental Results

We present experimental results obtained with **MultiComPoly**$_1$ on "*random*" decomposable instances of FDP_H. Here, we have randomly selected homogeneous polynomials $f = (f_1, \ldots, f_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$ of different degrees. Precisely, we have chosen $\deg(f_i) = 3$ if $i \leq \frac{u}{2}$, and $\deg(f_i) = 2$ if $i > \frac{u}{2}$. We then have randomly chosen homogeneous quadratic polynomials $g = (g_1, \ldots, g_n)$. The goal is then to decompose $h = (h_1, \ldots, h_u) = f \circ g$. Note that $\deg(h_i) = 6$ if $i \leq \frac{u}{2}$ and $\deg(h_i) = 4$ otherwise. Again, we have generated the instances using MAPLE, and we used FGb for computing the Gröbner bases. The polynomial ring is $\mathbb{F}_p[x_1, \ldots, x_n]$ with $p \approx 2^{32}$. We have quoted δ_{exp}, which is the smallest $\delta \geq 0$ s. t.:

$$\dim_{\mathbb{K}} \left(\mathrm{Vect}_{\mathbb{K}} \left(p \in D(\mathcal{I}_h) : x_n^{d + \delta} \mid \deg(p) = d_g \right) \right).$$

The theoretical value that you should obtain for a generic decomposition is :

$$\delta_{\mathrm{theo}} = \min_{\delta' \geq 0} \left\{ \frac{\sum_{i=1}^{u} u_i \cdot |M(\delta' + d - d_i)||M(d_{f_i} - 1)|}{|nM(\delta' + d)|} \geq 1 \right\}.$$

Finally, $\mathsf{T}_{\mathrm{New}}$ is the time needed for computing a d_g-Gröbner basis $D(\mathcal{I}_h) : x_n^{d + \delta_{\mathrm{exp}}}$.

u	n	d_{f_i}	δ_{exp}	δ_{theo}	$x_n^{d+\delta_{\mathrm{exp}}}$	$\mathsf{T}_{\mathrm{New}}$
6	6	2, 3	0	0	x_n^2	0.01 s.
8	8	2, 3	0	0	x_n^2	0.07 s.
10	10	2, 3	0	0	x_n^2	0.2 s.
12	12	2, 3	0	0	x_n^2	0.76 s.
14	14	2, 3	0	0	x_n^2	2.6 s.
16	16	2, 3	0	0	x_n^2	8.1 s.
18	18	2, 3	0	0	x_n^2	23.6 s.
20	20	2, 3	0	0	x_n^2	68.1 s.
4	8	2, 3	1	1	x_n^3	0.09 s.
5	9	2, 3	1	1	x_n^3	0.22 s.
5	10	2, 3	1	1	x_n^3	5.5 s.
6	12	2, 3	1	1	x_n^3	27.8 s.
7	14	2, 3	1	1	x_n^3	8.1 s.
8	16	2, 3	1	1	x_n^3	151.7 s.
9	18	2, 3	1	1	x_n^3	513.1 s.

We have quoted the minimum values of u and n for which the algorithm can return a decomposition. For a $([2, 3], 2)$ decomposition $(u = n)$, the algorithm will always return `Fail` if $n < 6$. We remark that a $([2, 3], 2)$ decomposition behaves as $(3, 2)$ decomposition. This is not surprising regarding how is constructed the matrix $A_{\widetilde{V}_\delta}$ defined in (4).

5. ACKNOWLEDGEMENT

We wish to thank Jaime Gutierrez and Joachim von zur Gathen for very helpful discussions on the decomposition problem. We also would like to thank the referees for their numerous comments which helped us to improve the quality of this paper.

6. REFERENCES

[1] V. S. Alagar and M. Thanh. *Fast Polynomial Decomposition Algorithms*. In Proc. EUROCAL85, Lecture Notes in Computer Science, vol. 204, pp. 150-153, Springer–Verlag, 1985.

[2] D. R. Barton and R. E. Zippel. *Polynomial decomposition algorithms.* J. Symb. Comp., 1, pp. 159–168, 1985.

[3] B. Buchberger. *An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal (German)*, PhD Thesis, University of Innsbruck, Math. Institute, Austria, 1965. (English Translation: J.S.C., Special Issue on Logic, Mathematics, and Computer Science: Interactions. Vol. 41 (3-4), pp 475-511, 2006).

[4] B. Buchberger. *Ein algorithmisches Kriterium fur die Lśsbarkeit eines algebraischen Gleichungssystems (An Algorithmical Criterion for the Solvability of Algebraic Systems of Equations)* Aequationes mathematicae 4/3, 1970, pp. 374-383. (English translation in: B. Buchberger, F. Winkler (eds.), Gröbner Bases

and Applications, Proc. of the International Conference “33 Years of Gröbner Bases”, 1998, RISC, Austria, London Mathematical Society Lecture Note Series, Vol. 251, Cambridge University Press, 1998, pp. 535 -545.)

[5] B. Buchberger. *Gröbner Bases : an Algorithmic Method in Polynomial Ideal Theory.* Recent trends in multidimensional systems theory. Reider ed. Bose, 1985.

[6] B. Buchberger, G.-E. Collins, and R. Loos. *Computer Algebra Symbolic and Algebraic Computation.* Springer-Verlag, second edition, 1982.

[7] E.-W. Chionh, X.-S. Gao, L.-Y. Shen. *Inherently Improper Surface Parametric Supports.* Computer Aided Geometric Design 23 (2006),pp. 629–639.

[8] D. A. Cox, J.B. Little, and D. O’Shea. *Ideals, Varieties, and Algorithms: an Introduction to Computational Algebraic Geometry and Commutative Algebra.* Undergraduate Texts in Mathematics. Springer-Verlag. New York, 1992.

[9] M. Dickerson. *The functional Decomposition of Polynomials.* Ph.D Thesis, TR 89-1023, Departement of Computer Science, Cornell University, Ithaca, NY, July 1989.

[10] M. Dickerson. *General Polynomial Decomposition and the s-1-decomposition are NP-hard.* International Journal of Foundations of Computer Science, 4:2 (1993), pp. 147–156.

[11] F. Dorey and G. Whaples. *Prime and composite polynomials.* J. Algebra,(28), pp. 88-101, 1974.

[12] J.-C. Faugère. *A New Efficient Algorithm for Computing Gröbner Basis without Reduction to Zero: F_5*. Proceedings of ISSAC, pp. 75–83. ACM press, July 2002.

[13] J.-C. Faugère, L. Perret. *Cryptanalysis of $2R^{-}$ schemes*. Advances in Cryptology – CRYPTO 2006, Lecture Notes in Computer Science, vol. 4117, pp. 357–372, Springer–Verlag, 2006.

[14] J.-C. Faugère, L. Perret. *An Efficient Algorithm for Decomposing Multivariate Polynomials and its Applications to Cryptography.* Special Issue of JSC, “Gröbner Bases techniques in Coding Theory and Cryptography”, on-line available.

[15] L. Goubin, and J. Patarin. *Asymmetric Cryptography with S-Boxes.* Information and Communication Security, First International Conference (ICICS’97), Lecture Notes in Computer Science vol. 1334, Springer–Verlag, pp. 369–380, 1997.

[16] L. Goubin, and J. Patarin. *Asymmetric Cryptography with S-Boxes* (Exentended version). `http://citeseer.ist.psu.edu/patarin97asymmetric.html`, 1997.

[17] J. Gutierrez, R. Rubio, J. von zur Gathen. *Multivariate Polynomial Decomposition*. Algebra in Engineering, Communication and Computing, 14 (1), pp. 11–31.

[18] J. Gutierrez, D. Sevilla. *Computation of Unirational fields.* J. Symb. Comput. 41(11), pp. 1222–1244, 2006.

[19] J. Gutierrez, R. Rubio, D. Sevilla. *On Multivariate Rational Function Decomposition*. J. Symb. Comput. 33(5), pp. 545–562, 2002.

[20] D. Kozen, and S. Landau. *Polynomial Decomposition Algorithms.* J. Symb. Comput. (7), pp. 445–456, 1989.

[21] J. F. Ritt. *Prime and Composite Polynomials.* Trans. Amer. Math. Soc., (23), pp 51-66, 1922.

[22] M. Sweedler. *Using Gröbner Bases to Determine the Algebraic and Transcendental Nature of Field Extensions: Return of the Killer Tag Variables.* Proc. AAECC, 66–75, 1993.

[23] J. Von zur Gathen. *Functional decomposition of polynomials: the tame case.* J. Symb. Comput. (9), pp. 281–299, 1990.

[24] J. Von zur Gathen. *Functional decomposition of polynomials: the wild case.* J. Symb. Comput. (10), pp. 437–452, 1990.

[25] S. M. Watt. *Functional Decomposition of Symbolic Polynomials.* In Proc. International Conference on Computational Sciences and its Applications, (ICCSA 2008), IEEE Computer Society, pp. 353–362.

[26] D.F. Ye, K.Y. Lam, Z.D. Dai. *Cryptanalysis of "2R" Schemes*, Advances in Cryptology – CRYPTO 1999, Lecture Notes in Computer Science, vol. 1666, Springer–Verlag, pp. 315–325, 1999.

[27] D.F. Ye, Z.D. Dai and K.Y. Lam. *Decomposing Attacks on Asymmetric Cryptography Based on Mapping Compositions*, Journal of Cryptology (14), pp. 137–150, 2001.