Cryptanalysis of $2R^-$ schemes

Jean-Charles Faugère[1] and Ludovic Perret[2]

[1] LIP6, 8 rue du Capitaine Scott, Paris 75015, France
Jean-Charles.Faugere@inria.fr
[2] UCL, Crypto Group, Microelectronic Laboratory
Place du Levant, 3
Louvain-la-Neuve, B 1348, Belgium
ludovic.perret@uclouvain.be

Abstract. In this paper, we study the security of $2R^-$ schemes [17, 18], which are the "minus variant" of two-round schemes. This variant consists in removing some of the n polynomials of the public key, and permits to thwart an attack described at Crypto'99 [25] against two-round schemes. Usually, the "minus variant" leads to a real strengthening of the considered schemes. We show here that this is actually not true for $2R^-$ schemes. We indeed propose an efficient algorithm for decomposing $2R^-$ schemes. For instance, we can remove up to $\left\lfloor \frac{n}{2} \right\rfloor$ equations and still be able to recover a decomposition in $O(n^{12})$. We provide experimental results illustrating the efficiency of our approach. In practice, we have been able to decompose $2R^-$ schemes in less than a handful of hours for most of the challenges proposed by the designers [18]. We believe that this result makes the principle of two-round schemes, including $2R^-$ schemes, useless.

Keywords : Cryptanalysis, Functional Decomposition Problem (FDP), Gröbner bases, F_5 algorithm.

1 Introduction

Last years a new kind of cryptanalysis has made its entrance in cryptography: the so-called *algebraic cryptanalysis*. A fundamental issue of this cryptanalysis consists in finding zeroes of algebraic systems. Gröbner bases, which are a fundamental tool of commutative algebra, constitute the most elegant and efficient way for solving this problem. They provide an algorithmic solution for solving several problems related to algebraic systems (some of them can be found in [1]). We present here a new application of Gröbner bases. More precisely, we propose a new algorithm for solving the *Functional Decomposition Problem* (FDP). The problem is as follows:
Functional Decomposition Problem (FDP)
Input : multivariate polynomials $h_1, \ldots, h_u$.
Find : – if any – multivariate polynomials $f_1, \ldots, f_u$, and $g_1, \ldots, g_n$, such that:

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

This problem is related to security of $2R^-$ schemes [17, 18].

1.1 Related Works

As stated by E. Biham [6], "the design of this scheme (2R) is unique as it uses techniques from symmetric ciphers in designing public key cryptosystems, while still claiming security based on relation to the difficulty of decomposing compositions of multivariate ... functions". Anyway, the security of 2R schemes has been already carefully investigated [6, 25, 26]. E. Biham proposed in [6] a successful cryptanalysis of 2R schemes with S-Boxes. This attack exploits the birthday paradox, but can be avoided by increasing the security parameters of 2R schemes [18]. At Crypto'99 [25], D.F. Ye, Z.D. Dai, and K.Y. Lam have presented a quite efficient method for solving the Functional Decomposition Problem. The security of 2R schemes is indeed related to this problem. To thwart this last attack, L. Goubin and J. Patarin have proposed [18] to use a general technique for repairing multivariate schemes, namely keeping secret some polynomials of the public key. The resulting schemes are called $2R^-$ schemes. Note that V. Carlier, H. Chabanne, and E. Dottax [9] have described a method for protecting the confidentiality of block ciphers design exploiting the principle of $2R^-$ schemes. Usually, the "minus modification" leads to a real strengthening of the considered schemes. For instance, C^* is broken [22] while C^{*--} is the basis of Sflash [10], the signature scheme recommended for low-cost smart cards by the European consortium Nessie[3]. Here, we show that $2R^-$ is not more secure than 2R.

1.2 Organization of the Paper and Main Results

The paper is organized as follows. We begin in Section 2 by introducing our notations and defining essential tools used in this paper, namely ideals, Gröbner bases, and several operations on ideals (sum, intersection, quotient, ...). Section 3 gives a brief review of one-round, 2R and $2R^-$ schemes. We also present the *Functional Decomposition problem* (FDP) in a more formal manner, which is at the basis of the security of 2R and $2R^-$ schemes. An algorithm for solving this problem efficiently would allow to decompose the public key of 2R and $2R^-$ schemes into two independent quadratic systems, making thereby the principle of these cryptosystems useless. In Section 4, we present a general algorithm for solving FDP. Our method is inspired on the algorithm of D.F. Ye, Z.D. Dai, and K.Y. Lam [25]. Note that their algorithm only works for particular instances of FDP, namely when $u = n$, or $u = n - 1$. Briefly, our algorithm works as follows. Let $(h_1, \ldots, h_u) = \big(f_1(g_1, \ldots, g_n), \ldots, f_u(g_1, \ldots, g_n)\big)$ be an instance of FDP. We first construct the ideal $\partial\mathcal{I}_h = \left\langle \frac{\partial h_i}{\partial x_j} : 1 \le i \le u, 1 \le j \le n \right\rangle$ generated by the partial derivatives of the h_is. We then show that for all $i, 1 \le i \le n, x_n^{d+1} g_i \in \partial\mathcal{I}_h$, for some $d \ge 0$. In most cases, this allows to recover a basis of the vector space $\mathcal{L}(g) = \mathrm{Vect}(g_1, \ldots, g_n)$ generated by $g_1, \ldots, g_n$. This is the most difficult part of our algorithm. The f_is being indeed recovered from the knowledge of $\mathcal{L}(g)$ by solving a linear system. The complexity of this algorithm depends on the ratio n/u. For example, our algorithm runs in $O(n^{12})$, if $n/u < 1/2$. More generally, we provide a global analysis of the theoretical complexity of our method. As a side effect, we give several insights into the theoretical behavior of the algorithm of D.F. Ye, Z.D. Dai, and K.Y. Lam. We conclude this section by providing experimental results illustrating the efficiency of our approach. We have been able to solve in few hours instances of FDP used in $2R^-$ schemes for most of the challenges proposed in [18].

2 Preliminaries

Throughout this paper, we denote by $\mathbb{K}[x_1, \ldots, x_n]$ the polynomial ring in the n indeterminate $x_1, \ldots, x_n$ over a finite field $\mathbb{K}$ with $q = p^r$ elements (p a prime, and $r \ge 1$). The set of polynomials $p_1, \ldots, p_s$ of $\mathbb{K}[x_1, \ldots, x_n]$ can be regarded as a mapping $\mathbb{K}^n \to \mathbb{K}^s$:

$$(v_1, \ldots, v_n) \mapsto \big(p_1(v_1, \ldots, v_n), \ldots, p_s(v_1, \ldots, v_n)\big).$$

We will call these polynomials *components*. We will also denote by $\mathcal{I} = \langle p_1, \ldots, p_s \rangle = \{\sum_{k=1}^{s} p_k u_k \colon u_1, \ldots, u_s \in \mathbb{K}[x_1, \ldots, x_n]\}$ the *ideal generated* by $p_1, \ldots, p_s$. We define now essential notions used in this paper. For a more thorough introduction to these tools, we refer to classical books on commutative algebra, such as [1, 11]. Most of the results presented in this part are well known in commutative algebra, and thus given without proofs. For these proofs, we also refer to [1, 11]. The reader already familiar with Gröbner bases and quotient ideals can skip this part.

[3] `https://www.cosic.esat.kuleuven.be/nessie/deliverables/decision-final.pdf`.

2.1 Gröbner bases

Informally, a Gröbner basis of an ideal is a generatring set of this ideal with "good" algorithmic properties. These bases are defined with respect to *monomial orders*. Here, we will use the lexicographic (LEX) and degree reverse lexicographical (DRL) orders, which are definedas follows:

Definition 1. *Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{N}^n$. Then:*
– $x_1^{\alpha_1} \cdots x_n^{\alpha_n} \prec_{LEX} x_1^{\beta_1} \cdots x_n^{\beta_n}$, if the left-most nonzero entry of the vector $\alpha - \beta$ is positive.
– $x_1^{\alpha_1} \cdots x_n^{\alpha_n} \prec_{DRL} x_1^{\beta_1} \cdots x_n^{\beta_n}$, if $\sum_{i=1}^n \alpha_i > \sum_{i=1}^n \beta_i$, or $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i$ and the right-most nonzero entry of $\alpha - \beta$ is negative.

To define Gröbner bases, we have to introduce the following definitions.

Definition 2. *For any n-uple $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, we denote by x^α the* **monomial** *$x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. We define the* total degree *of this monomial by the sum $\sum_{i=1}^n \alpha_i$. The* **leading monomial** *of a polynomial $f \in \mathbb{K}[x_1, \ldots, x_n]$ is the largest monomial – w.r.t some monomial ordering $\prec$ – among the monomials of f. This leading monomial will be denoted by $\mathrm{LM}(f, \prec)$. The* **leading coefficient** *of f, denoted by $\mathrm{LC}(f, \prec)$, is the coefficient of $\mathrm{LM}(f, \prec)$ in f. The* **degree** *of f – denoted deg(f) – is the total degree of $\mathrm{LM}(f, \prec)$. Finally, the* **maximal total degree** *of f is the maximal total degree of the monomials occurring in f.*

We are now ready to define one of the main objects of this paper. Indeed:

Definition 3. *A set of polynomials G is a* **Gröbner basis** *– w.r.t. a monomial ordering $\prec$ – of an ideal $\mathcal{I}$ in $\mathbb{K}[x_1, \ldots, x_n]$, if for all $f \in \mathcal{I}$ there exists $g \in G$ such that $\mathrm{LM}(g, \prec)$ divides $\mathrm{LM}(f, \prec)$. This Gröbner basis is called* **reduced** *if, for all $g \in G$, $\mathrm{LC}(g, \prec) = 1$, and any monomial of $g \in G$ is not divisible by any element of $\mathrm{LM}(G \backslash \{g\}, \prec)$. Let G be Gröbner basis – w.r.t. a monomial ordering $\prec$ – of an ideal $\mathcal{I}$ in $\mathbb{K}[x_1, \ldots, x_n]$, and d be a positive integer. We call d-***Gröbner basis** *(or* **truncated Gröbner basis***) of an homogeneous ideal $\mathcal{I}$ the set:*

$$\{g \in G : deg(g) = d\}.$$

A Gröbner basis of a given ideal is not unique in general. The reduced Gröbner basis allows to achieve uniqueness. A reduced Gröbner basis can be obtained from a Gröbner basis in polynomial-time. Gröbner bases are a fundamental tool to study algebraic systems in theory and practice. They provide an algorithmic solution for solving several problems related to polynomial systems (some of them can be found in [1]). The historical method for computing Gröbner bases is Buchberger's algorithm [8, 7]. Recently, more efficient algorithms have been proposed. To date, F_5 [13] is the most efficient for computing Gröbner bases (a brief description of this algorithm is given in Appendix A). Here we will concentrate on Gröbner bases w.r.t. lexicographical and degree reverse lexicographical orders.

LEX and DRL Gröbner bases
Lexicographical Gröbner bases (LEX Gröbner bases) offer a way for eliminating variables.

Theorem 1 (Elimination Theorem). *Let $\mathcal{I}$ be an ideal in $\mathbb{K}[x_1, \ldots, x_n]$, and $k \in \{1, \ldots, n\}$. If G is a LEX Gröbner basis of $\mathcal{I}$, then $G \cap \mathbb{K}[x_{k+1}, \ldots, x_n]$ is a Gröbner basis of $\mathcal{I} \cap \mathbb{K}[x_{k+1}, \ldots, x_n]$.*

The shape of degree reverse lexicographical Gröbner bases (DRL Gröbner bases) is much more complicated. However, DRL Gröbner bases have several interesting properties. For instance, the polynomials of lowest degree of an ideal $\mathcal{I}$ appear in a DRL Gröbner bases of this ideal. More precisely:

Theorem 2. *Let $\mathcal{I} \subset \mathbb{K}[x_1, \ldots, x_n]$, $d = min\{deg(f) : f \in \mathcal{I}\}$, and G be a DRL Gröbner basis of $\mathcal{I}$. Then:*

$$\mathrm{Vect}\big(g \in G : deg(g) = d\big) = \mathrm{Vect}\big(g \in \mathcal{I} : deg(g) = d\big).$$

Proof. A proof of this theorem can be found in [3]. □

We should mention that the variable x_n has a special role for the DRL order.

Lemma 1. *Let $f \in \mathbb{K}[x_1, \ldots, x_n]$, and m be a positive integer. Then:*

$$x_n^m | f \iff x_n^m | \mathrm{LM}(f, \prec_{DRL}).$$

Sum, Intersection, and Quotient of Ideals
We now go over the definitions of several operations on ideals.

Definition 4. *Let $\mathcal{I}$ and $\mathcal{J}$ be ideals in $\mathbb{K}[x_1, \ldots, x_n]$. Then:*
– the **sum** *of $\mathcal{I}$ and $\mathcal{J}$, noted $\mathcal{I} + \mathcal{J}$, is the $\mathcal{I} + \mathcal{J} = \{f + g : f \in \mathcal{I}$ and $g \in \mathcal{J}\}$.*
– the **intersection** *of $\mathcal{I}$ and $\mathcal{J}$, is defined as $\mathcal{I} \cap \mathcal{J} = \{f \in \mathbb{K}[x_1, \ldots, x_n] : f \in \mathcal{I}$ and $f \in \mathcal{J}\}$.*
– $\mathcal{I} + \mathcal{J}$ and $\mathcal{I} \cap \mathcal{J}$ are ideals.

Given two ideals and their generators, we would like to compute a set of generators for the intersection. This is actually much more delicate than the analogous problem for sums, which is straightforward. Indeed, $\mathcal{I} = \langle p_1, \ldots, p_s \rangle + \mathcal{J} = \langle g_1, \ldots, g_r \rangle = \langle p_1, \ldots, p_s, g_1, \ldots, g_r \rangle$. The following result permits to solve the problem for intersections.

Theorem 3. *Let $\mathcal{I}, \mathcal{J}$ be ideals in $\mathbb{K}[x_1, \ldots, x_n]$, and t be a new variable. Then:*

$$\mathcal{I} \cap \mathcal{J} = \big(t \cdot \mathcal{I} + (1-t) \cdot \mathcal{J}\big) \cap \mathbb{K}[x_1, \ldots, x_n],$$

where $t \cdot \mathcal{I} = \{t \cdot h : h \in \mathcal{I}\}$, and $(1-t) \cdot \mathcal{J} = \{(1-t) \cdot h : h \in \mathcal{J}\}$ are in $\mathbb{K}[t, x_1, \ldots, x_n]$.

This result, together with the Elimination Theorem (i.e. Theorem 1), provide a method for computing intersections of ideals. Given ideals $\mathcal{I} = \langle p_1, \ldots, p_s \rangle$ and $\mathcal{J} = \langle g_1, \ldots, g_r \rangle$ in $\mathbb{K}[x_1, \ldots, x_n]$, we consider the ideal $\langle t \cdot p_1, \ldots, t \cdot p_s, (1-t) \cdot g_1, \ldots, (1-t) \cdot g_r \rangle \subset \mathbb{K}[t, x_1, \ldots, x_n]$. Those elements of a LEX Gröbner basis (with $t \succ_{LEX} x_1 \succ_{LEX} \cdots \succ_{LEX} x_n$) that do not contain the variable t will exactly form a Gröbner basis for $\mathcal{I} \cap \mathcal{J}$.

Definition 5. *Let $\mathcal{I}$ and $\mathcal{J}$ be ideals in $\mathbb{K}[x_1, \ldots, x_n]$. The* **ideal quotient** *of $\mathcal{I}$ by $\mathcal{J}$, denoted $\mathcal{I} : \mathcal{J}$, is the set*

$$\mathcal{I} : \mathcal{J} = \{f \in \mathbb{K}[x_1, \ldots, x_n] : fg \in \mathcal{I}, \text{ for all } g \in \mathcal{J}\}.$$

The following proposition relates the quotient operation to the sum and intersection operations.

Proposition 1. *Let $\mathcal{I}$, and $\{\mathcal{I}_k\}_{1 \le k \le r}$ be ideals in $\mathbb{K}[x_1, \ldots, x_n]$. Then:*

$$\text{i) } \left(\bigcap_{k=1}^r \mathcal{I}_k\right) : \mathcal{I} = \bigcap_{k=1}^r (\mathcal{I}_k : \mathcal{I})$$
$$\text{ii) } \mathcal{I} : \left(\sum_{k=1}^r \mathcal{I}_k\right) = \bigcap_{k=1}^r (\mathcal{I} : \mathcal{I}_k)$$

If f is a polynomial and $\mathcal{I}$ an ideal, we shall write $\mathcal{I} : f$ instead of $\mathcal{I} : \langle f \rangle$. A special case of ii) is:

$$\mathcal{I} : \langle f_1, \ldots, f_r \rangle = \bigcap_{k=1}^r (\mathcal{I} : f_k).$$

We now address the question of computing generators of the ideal quotient $\mathcal{I} : \mathcal{J}$. The following observation is crucial:

Theorem 4. *Let $\mathcal{I}$ be an ideal in $\mathbb{K}[x_1, \ldots, x_n]$, and $f \in \mathbb{K}[x_1, \ldots, x_n]$. If $\langle g_1, \ldots, g_p \rangle = \mathcal{I} \cap \langle f \rangle$, then*

$$\langle g_1/f, \ldots, g_p/f \rangle = \mathcal{I} : f.$$

In order to construct a basis of an ideal quotient, we proceed as follows. Given ideals $\mathcal{I} = \langle p_1, \ldots, p_s \rangle$ and $\mathcal{J} = \langle g_1, \ldots, g_r \rangle$ in $\mathbb{K}[x_1, \ldots, x_n]$, we compute a basis for the intersections $\mathcal{I} \cap \langle g_1 \rangle, \ldots, \mathcal{I} \cap \langle g_n \rangle$ by using the above described method. For each i, we divide by f each element of a basis of $\mathcal{I} \cap \langle g_i \rangle$. This leads to a basis for $\mathcal{I} : g_i$. We then obtain a basis for $\mathcal{I} : \mathcal{J}$ by computing the intersections $\bigcap_{k=1}^r (\mathcal{I} : g_i)$.

3 $2R^-$ schemes

In [20], T. Matsumoto and H. Imai proposed one of the first examples of PKCs using compositions of multivariate polynomials. The public key of one of them, called C* ([21]), represented by "$t \circ \psi \circ s$", where t, s are two secret linear mappings over $GF(2)^n$, and ψ is the multivariate representation of $GF(2^n) \to GF(2^n), x \mapsto x^{1+2^\theta}$. This scheme has been broken by J. Patarin at Crypto'95 [22].
One-round schemes [17, 18] are generalizations of C*. The public key of these schemes is indeed of the form "$t \circ \psi \circ s$", where t, s are two affine mappings over $\mathbb{K}^n$, and a $\psi : \mathbb{K}^n \to \mathbb{K}^n$ is a bijective mapping given by n multivariate polynomials of degree two. J. Patarin and L. Goubin [17, 18] propose several constructions for ψ:

1. S-box functions: $(a_1, \ldots, a_n) \mapsto$

$$\big(S_1(a_1 \ldots, a_{n_1}), S_2(a_{n_1+1} \ldots, a_{n_1+n_2}), \ldots, S_b(a_{n_1+n_2+\cdots+n_{d-1}+1}, \ldots, a_n)\big),$$

where $n = \sum_i n_i$, and each $S_i : K^{n_i} \to K^{n_i}$ is quadratic.
2. Triangular functions:

$$(a_1, \ldots, a_n) \mapsto \big(a_1, a_2 + q_1(a_1), a_3 + q_2(a_1, a_2, a_3), \ldots, a_n + q_{n-1}(a_1, \ldots, a_{n-1})\big),$$

where each q_i is quadratic.
3. Combinations of S-box and triangular functions.

They showed that all these constructions are insecure [17, 18]. To circumvent attacks, they introduce *two-round schemes* whose public key is the composition of two one-round schemes. The secret key of two-round schemes consists of:

Three affine bijections $r, s, t : K^n \to K^n$.
Two applications $\psi, \phi : K^n \to K^n$, given by n quadratic polynomials.

The public key is composed of n polynomials $p_1, \ldots, p_n$ of total degree 4 describing:

$$p = t \circ \psi \circ s \circ \phi \circ r, K^n \to K^n.$$

When all the polynomials are given, this scheme is called *2R scheme*. If only some of them are given, it is called *2R$^-$ scheme*. The public-key part of the computation is merely an application of the mapping p (for encrypting a message, or checking the validity of a signature). For the secret-key computations, we need to invert the mappings ψ and ϕ. The authors then propose to choose the mappings among the constructions 1, 2, 3 described above and also:

4. C* functions: monomials over an extension of degree n over K,
5. D* functions [16].

In [17, 18], it has been proved that when ψ is chosen in the classes 2. and 4., then the resulting 2R scheme is weak. It is not clear that a similar result holds for $2R^-$ schemes.
Anyway, does composing two weak one-round schemes leads to a secure scheme ? The answer is closely related to the difficulty of the following problem:
Functional Decomposition Problem (FDP)
Input : $h = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$.
Find : – if any – $f = (f_1, \ldots, f_u) \neq h \in \mathbb{K}[x_1, \ldots, x_n]^u$, and $g = (g_1, \ldots, g_n) \in \mathbb{K}[x_1, \ldots, x_n]^n$, such that:

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

noted $h(x) = (f \circ g)(x)$ hereafter, where $x = (x_1, \ldots, x_n)$.
During the last years several results have been obtained on the univariate polynomial decomposition area [15, 23, 24]. However, multivariate decomposition problem has not been studied so much. Particular instances (multi-univariate,...)

of FDP have been investigated in [23, 19]. In [12], M. Dickerson provided several insights into the theoretical complexity of FDP. However, this kind of results solely guarantees the difficulty of the worst-case. In the cryptographic context, D.F. Ye, Z.D. Dai, and K.Y. Lam presented in [25, 26] a quite efficient method for solving instances of FDP used in 2R. Note that their method only works when $u = n$, or $u = n - 1$ [26]. To the best of our knowledge, there exists no previously known algorithm for solving FDP when $u < n - 1$. An efficient method for solving FDP in this case would permit to decompose $2R^-$ schemes into two independent schemes given by quadratic polynomials. To break these schemes, we then would only have to solve two quadratic systems. As mentioned by J. Patarin and L. Goubin [18], this would make the principle of two-round schemes, including $2R^-$, useless.

4 A general algorithm for solving FDP

In this part, we present a new algorithm for solving FDP. Our approach is inspired on the works of D.F. Ye, Z.D. Dai, and K.Y. Lam [25, 26]. According to these authors, we can restrict our attention to homogeneous instances of FDP [25]. The *homogenization* of a polynomial $p \in \mathbb{K}[x_1, \ldots, x_n]$, denoted p^*, is defined by $p^*(x_0, x_1, \ldots, x_n) = x_0^{deg(p)} p(x_1/x_0, \ldots, x_n/x_0)$, where x_0 is a new variable. For any mapping $f : \mathbb{K}^n \to \mathbb{K}^u$, given by the polynomials $f_1, \ldots, f_u$, we define its *homogenization* by $f^* = (x_0^{deg(f)}, f_1^*, \ldots, f_n^*)$. The *dehomogenization* of f^* is then $f = \big(f_1^*(1, x_1, \ldots, x_n), \ldots, f_n^*(1, x_1, \ldots, x_n)\big)$. We have:

Lemma 2 ([25]). *Let $f : \mathbb{K}^n \to \mathbb{K}^u$ and $g : \mathbb{K}^n \to \mathbb{K}^n$ be two mappings, then:*

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

Note 1. In [25], it is stated that this lemma is correct only if $deg(f)deg(g) > |K|$. We no longer need this condition over $\mathbb{K}[x_1, \ldots, x_n]$.

Thus, if we can decompose $h^* = f^* \circ g^*$, then a decomposition of $h = f \circ g$ is simply obtained by dehomogenization of f^* and g^*[25]. Now, we assume that $f : \mathbb{K}^n \to \mathbb{K}^u$ and $g : \mathbb{K}^n \to \mathbb{K}^n$ are two homogeneous functions of degree two. Finally, let $h = f \circ g$, and $\{h_i\}_{1 \le i \le u}, \{f_i\}_{1 \le i \le u}, \{g_i\}_{1 \le i \le n}$ be the components of h, f, g respectively.

4.1 Description of the algorithm

The aim of our algorithm is to find the vector space $\mathcal{L}(g) = \mathrm{Vect}(g_1, \ldots, g_n)$ generated by $g_1, \ldots, g_n$. More precisely, this vector space will be recovered from a DRL Gröbner basis of a suitable ideal. Note that the knowledge of $\mathcal{L}(g)$ is sufficient for decomposing h. Indeed, any bijective linear combination A of the g_is leads to a decomposition of h since:

$$h = (f \circ A^{-1}) \circ (A \circ g).$$

Let us first assume that we know the vector space $\mathcal{L}(g)$. For all $i, 1 \le i \le u$:

$$\begin{aligned} f_i &= \textstyle\sum_{1 \le k, \ell \le n} f_{k,\ell}^{(i)} x_k x_\ell \in \mathbb{K}[x_1, \ldots, x_n], \\ g_i &= \textstyle\sum_{1 \le k, \ell \le n} g_{k,\ell}^{(i)} x_k x_\ell \in \mathbb{K}[x_1, \ldots, x_n]. \end{aligned}$$

Therefore, for all $i, 1 \le i \le u$:

$$h_i = f_i(g_1, \ldots, g_n) = \sum_{1 \le k, \ell \le n} f_{k,\ell}^{(i)} g_k g_\ell. \tag{1}$$

By comparing the coefficients in the right-most and left-most parts of these equalities, we obtain a linear system of $O(u\mathrm{C}_{n+2}^2)$ equations in the $u\mathrm{C}_{n+2}^2$ unknown coefficients of the f_is. It seems difficult to rigorously evaluate the rank of this linear system, a question that has been avoided in the previous works on FDP [25, 26]. However, it is very likely that this linear system is of full rank when the f_is are dense polynomials. For the instances of FDP used in

2R$^-$ schemes, we experimentally only obtain linear systems of full rank. The difficult part is actually to determine the vector space $\mathcal{L}(g)$. For this, we observe that:

$$\frac{\partial h_i}{\partial x_j} = \sum_{1 \le k,\ell \le n} f_{k,\ell} \left(\frac{\partial g_k}{\partial x_j} g_\ell + g_k \frac{\partial g_\ell}{\partial x_j} \right). \tag{2}$$

The polynomials $g_1, \ldots, g_n$ being of degree two, their partial derivatives are of degree one. Hence:

$$\partial \mathcal{I}_h = \left\langle \frac{\partial h_i}{\partial x_j} : 1 \le i \le u, 1 \le j \le n \right\rangle \subseteq \langle x_k g_\ell \rangle_{1 \le k,\ell \le n} = \mathcal{V}.$$

This ideal $\partial \mathcal{I}_h$ usually provides enough information for recovering the polynomials $g_1, \ldots, g_n$.

Theorem 5. *Let $M(d)$ be the set of monomials of degree $d \ge 0$ in $x_1, \ldots, x_n$, and*

$$V_d = \mathrm{Vect}\left(m g_k : m \in M(d+1), \text{ and } 1 \le k \le n\right),$$
$$\tilde{V}_d = \mathrm{Vect}\left(\left\{ m \frac{\partial h_i}{\partial x_j} : m \in M(d), 1 \le i \le u, \text{ and } 1 \le j \le n \right\} \right).$$

Then, for all $i, 1 \le i \le n$:

$$x_n^{d+1} g_i \in \partial \mathcal{I}_h, \text{ if } \dim(\tilde{V}_d) \ge n|M(d+1)|,$$

where $\dim(\tilde{V}_d)$ is the dimension of $\tilde{V}_d$ as a vector space over V_d.

Proof. We first study the case $d = 0$. Let $\tilde{V} = \tilde{V}_0$ be the linear space generated by the partial derivatives of the h_is, i.e.:

$$\tilde{V}_0 = \tilde{V} = \mathrm{Vect}\left(\left\{ \frac{\partial h_i}{\partial x_j} \right\}_{1 \le j \le n}^{1 \le i \le u} \right) \subset \partial \mathcal{I}_h.$$

According to (2), each element of $\tilde{V}$ can be written as a sum of $\{x_k g_\ell\}_{1 \le k,\ell \le n}$. Now let $A_{\tilde{V}} \in \mathcal{M}_{n^2 \times n^2}(\mathbb{K})$ be a matrix associated to the linear transformation $\mathrm{Vect}\big(\{x_k g_\ell\}_{1 \le k,\ell \le n}\big) \mapsto \tilde{V}$. For some basis:

$$A_{\tilde{V}} = \begin{array}{c} \\ \frac{\partial h_1}{\partial x_1} \\ \vdots \\ \frac{\partial h_1}{\partial x_n} \\ \vdots \\ \frac{\partial h_i}{\partial x_j} \\ \vdots \\ \frac{\partial h_n}{\partial x_1} \\ \vdots \\ \frac{\partial h_n}{\partial x_n} \end{array} \begin{array}{c} \begin{array}{ccccccccc} x_1 g_1 & \cdots & x_n g_1 & \cdots & x_k g_\ell & \cdots & x_1 g_n & \cdots & x_n g_n \end{array} \\ \left(\begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} \right) \end{array}$$

One can see at once that the $x_n g_i$s lie in $\tilde{V}$ if the number of linearly independent rows of this matrix is at least equal to its number of columns. That is, $x_n g_i \in \partial \mathcal{I}_h$, for all $i, 1 \le i \le n$, if:

$$\dim(\tilde{V}) \ge n|M(1)| = n^2.$$

Observe that $\dim(\tilde{V})$ is upper-bounded by un. Thus, $\dim(\tilde{V}) \ge n|M(1)| = n^2$ only holds if $u = n$. This explains why the method proposed in [25, 26] is limited to 2R schemes. To circumvent this problem, we have to consider a vector space of higher dimension. This is the motivation for considering:

$$\tilde{V}_d = \mathrm{Vect}\big(\left\{ m \frac{\partial h_i}{\partial x_j} : m \in M(d), 1 \le i \le u, \text{ and } 1 \le j \le n \right\} \big).$$

From (2), we deduce that each polynomial of $\tilde{V}_d$ can be written as a sum of elements of:

$$V_d = \mathrm{Vect}\left(mg_k : m \in M(d+1), \text{ and } 1 \le k \le n\right).$$

Let then $A_{\tilde{V}_d}$ be a matrix associated to $V_d \mapsto \tilde{V}_d$. For some basis:

$$A_{\tilde{V}_d} = \begin{array}{c} \\ \vdots \\ \vdots \\ m\frac{\partial h_i}{\partial x_j} \\ \vdots \\ \vdots \end{array} \begin{array}{c} \cdots \quad \cdots \quad mg_k \quad \cdots \quad \cdots \\ \left(\begin{array}{ccc} & \cdots & \\ & \cdots & \\ & \cdots & \\ & \cdots & \\ & \cdots & \end{array} \right) \end{array}$$

Thus, $x_n^{d+1} g_i \in \tilde{V} \subset \partial\mathcal{I}_h,$ for all $i, 1 \le i \le n$, if $\dim(\tilde{V}_d)$ is at least equal to the number of columns of $A_{\tilde{V}_d}$. That is, if $\dim(\tilde{V}_d) \ge n|M(d+1)|$. □

Remark 1. *At the end of this part, we will provide an explicit value of d in function of the ratio n/u.*

According to Theorem 5, the polynomials g_is are contained, up to some power of x_n, in $\partial\mathcal{I}_h$. Therefore, the quotient of this ideal by a suitable power of x_n contains the polynomials $g_1, \ldots, g_n$.

Corollary 1. *Using the same notations as in Theorem 5. If $\dim(\tilde{V}_d) \ge n|M(d+1)|$, then:*

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

Proof. The proof of this corollary is obviously deduced from Theorem 5, and very definition of the quotient. □

Thus each element of $\mathcal{L}(g)$ is included in $\partial\mathcal{I}_h : (x_n^{d+1})$. Let then G be a (reduced) DRL Gröbner basis of this ideal. It is then natural to consider the set $B_g = \mathrm{Vect}\big(g \in G : deg(g) = 2\big)$, since according to Theorem 2:

$$\mathcal{L}(g) = B_g, \text{if } \#B_g = n, \text{ and } min\big(deg(g) : g \in G\big) = 2.$$

If these conditions are not fulfilled, then one can not recover efficiently $\mathcal{L}(g)$ from B_g. Observe that the condition $\#B_g = n$ implies that there exists a unique decomposition (up to bijective linear combinations). To get away with these problems, we can apply several heuristics such as computing $\partial\mathcal{I}_h : (x_1^{d+1}), \ldots, \partial\mathcal{I}_h : (x_{n-1}^{d+1})$. In practice, it has been always sufficient to compute $\partial\mathcal{I}_h : (x_n^d)$, for a suitable d (i.e. $\dim(\tilde{V}_d) \ge n|M(d+1)|$).

4.2 The algorithm $\mathbf{Algo_{FDP}}$

We describe now our algorithm for general instances of FDP, i.e. we no longer suppose here that h is given by homogeneous polynomials.

Algo$_{\text{FDP}}$
Input: $h = f \circ g : \mathbb{K}^n \to \mathbb{K}^u$, given by u polynomials $h_1, \ldots, h_u \in \mathbb{K}[x_1, \ldots, x_n]$ of degree 4
Output : $f'_1, \ldots, f'_u, g'_1, \ldots, g'_n$, such that $\big(h_1, \ldots, h_u\big) = \big(f'_1(g'_1, \ldots, g'_n), \ldots, f'_u(g'_1, \ldots, g'_n)\big)$
$h_0^*(x_0, x_1, \ldots, x_n) \leftarrow x_0^4$
$h_i^*(x_0, x_1, \ldots, x_n) \leftarrow x_0^4 h_i(x_1/x_0, \ldots, x_n/x_0)$, for all $i, 1 \leq i \leq u$
$\partial\mathcal{I}_h^* \leftarrow \left\langle \frac{\partial h_i^*}{\partial x_j} : 0 \leq i \leq u, 0 \leq j \leq n \right\rangle$
Let d be the smallest integer such that $\dim(\tilde{V}_d^*) \geq n|M(d+1)|$, with:
$\tilde{V}_d^* = \mathrm{Vect}\left(\left\{m \frac{\partial h_i^*}{\partial x_j} : m \in M(d), 0 \leq i \leq u, \text{ and } 0 \leq j \leq n\right\}\right)$.
Compute a reduced 2-DRL Gröbner basis G of $\partial\mathcal{I}_h^* : (x_n^{d+1})$
$B_{g^*} \leftarrow \big\{g^* \in G, deg(g^*) = 2\big\}$
If $\#B_{g^*} \neq n+1$ **or** $min\big(deg(g) : g \in G\big) \neq 2$ **then** Return `Fail`
Recover a basis B_{f^*} of $\mathrm{Vect}(f^*)$ by solving the system of linear equations given by (1)
Return $\big\{g^*(1, x_1, \ldots, x_n) \in B_{g^*}\big\}$ and $\big\{f^*(1, x_1, \ldots, x_n) \in B_{f^*}\big\}$

Remark 2. *In practice, our algorithm never returned* `Fail` *for instances of FDP used in $2R^-$.*

Theorem 6. *Let $g_0^*(x_0, x_1, \ldots, x_n) = x_0^2$, and $g_i^*(x_0, x_1, \ldots, x_n) = x_0^2 g_i(x_1/x_0, \ldots, x_n/x_0)$, for all $i, 1 \leq i \leq n$. Moreover, let $M(d)$ be the set of monomials of degree $d \geq 0$ in $x_0, x_1, \ldots, x_n$, and*

$$V_d^* = \mathrm{Vect}\left(m g_k^* : m \in M(d+1), \text{ and } 0 \leq k \leq n\right),$$
$$\tilde{V}_d^* = \mathrm{Vect}\left(\left\{m \frac{\partial h_i^*}{\partial x_j} : m \in M(d), 0 \leq i \leq u, \text{ and } 0 \leq j \leq n\right\}\right).$$

Algo$_{\text{FDP}}$ *returns a solution of FDP (and not* `Fail`*) in:*

$$O(n^{3(d+3)}),$$

where d is the smallest integer such that $dim(\tilde{V}_d^*) \geq (n+1)|M(d+1)|$.

Proof. Let us suppose that our algorithm returns a solution (and not `Fail`). According to Corollary 1, we know that for all $i, 0 \leq i \leq n, g_i^* \in \partial\mathcal{I}_h^* : (x_n^{d+1})$. The complexity of Algo$_{\text{FDP}}$ is then dominated by the cost of computing a reduced DRL Gröbner basis G of $\partial\mathcal{I}_h^* : (x_n^{d+1})$. This step can be done as explained in Section 2. However, an alternative method can be used in this particular situation. This is due to the particular role of x_n in a DRL order. From Lemma 1, we know that if x_n^{d+1} divides the leading monomial of a polynomial, then it also divides the entire polynomial. Thus, we can restrict our attention to polynomials of a DRL Gröbner Bases G' of $\partial\mathcal{I}_h^*$ whose leading monomials contain x_n^{d+1}. One can see directly that:

$$\big(g \in G : deg(g) = 2\big) = \left(\frac{g'}{x_n^{d+1}} : g' \in G', \text{and } x_n^{d+1} | \mathrm{LM}(g', \prec_{DRL})\right).$$

More precisely, it is sufficient to compute a reduced $(d+3)$-DRL Gröbner basis of $\partial\mathcal{I}_h^*$. According to Appendix A, this can be done with the F_5 algorithm in $O(n^{3(d+3)})$. From a practical point of view, the two methods proposed for computing G are similar. But the last one is more suitable for evaluating the complexity. □

Remark 3. *It should be noticed that our algorithm can easily be adapted for polynomials f of degree greater that 2.*

Comparison with previous approach
In short, our method can be viewed as a generalization of the approach of D.F. Ye, Z.D. Dai, and K.Y. Lam [25, 26]. When $u = n$, it is sufficient to consider the ideal $\partial\mathcal{I}_h^* : (x_n^1)$ for recovering $\mathcal{L}(g)$. This is a simplified description of the method described in [25, 26]. When $u < n$, $\partial\mathcal{I}_h^* : (x_n^1)$ no longer provides enough information for recovering $\mathcal{L}(g)$. To overcome this difficulty, we proposed here to consider ideals of the form $\partial\mathcal{I}_h^* : (x_n^{d+1})$. We then proved that $\mathcal{L}(g)$ is contained in this ideal as soon as d is sufficiently large.

It is important to know the exact value of the parameter d. This value can be lower-bounded in fonction of the ratio n/u. For this, we observe that $(n+1)|M(d+1)| = (n+1)\mathrm{C}_{n+1+d}^{d+1}$ and $\dim(\tilde{V}_d^*)$ is very likely to be equal $(u+1)(n+1)\mathrm{C}_{n+d}^{d}$. We then obtain that d should verify:

$$d \geq \frac{n}{u} - 1.$$

For instance, if the number of equations removed (i.e. $n-u$) is smaller than $\lfloor \frac{n}{2} \rfloor$, this yields a complexity of $O(n^{12})$, and $O(n^9)$ if $u = n$. We will show now that this approximation is perfectly coherent with our experimental results.

4.3 Experimental results

Generation of the instances
We have only considered instances $h = f \circ g$ of FDP admitting a solution. We constructed these instances in the following way:
– $f = t \circ \psi \circ s$ and $g = \phi \circ r$, with $r, s, t \circ \psi \circ s : K^n \rightarrow K^n$ are random affine bijections, and $\psi, \phi : K^n \rightarrow K^n$ are S-box functions contructed as explained in Section 3. We then remove $r \geq 0$ polynomials of h.

Programming language – Workstation
The experimental results have been obtained with a Xeon bi-processor 3.2 Ghz, with 6 Gb of Ram. The instances of FDP have been generated using the Maple software. We used our own implementation (in language C) of F_5 for computing truncated Gröbner bases.

Table Notations
The following notations are used in the table below:
– n, the number of variables,
– b, the number of blocks (as defined in Section 3),
– n_i, the number of variables in each block (see Section 3),
– q, the size of the field,
– r, the number of polynomials removed,
– $d_{theo} = \lceil \frac{n}{u} - 1 \rceil$, the predicted (see 4.2) value of d for which $\mathrm{Algo}_{\mathrm{FDP}}$ returns a solution
– d_{real}, the real value of d for which $\mathrm{Algo}_{\mathrm{FDP}}$ returns a solution
– T, the total time taken by our algorithm,
– $\sqrt{q^n}$, the current security bound [18, 6] for $2\mathrm{R}^-$ schemes.

Practical Results
Let us now present results obtained with our algorithm.

n	b	n_i	r	q	d_{theo}	d_{real}	T	$\sqrt{q^n}$
8	4	2	0	65521	0	0	0.0 s.	
8	4	2	4	65521	1	1	0.0 s.	$\approx 2^{64}$
8	4	2	5	65521	2	2	0.3 s.	$\approx 2^{64}$
8	4	2	6	65521	3	3	1.9 s.	$\approx 2^{64}$
10	5	2	5	65521	1	1	0.2 s.	$\approx 2^{80}$
10	5	2	6	65521	2	2	3.2 s.	$\approx 2^{80}$
10	5	2	7	65521	3	3	21.4 s.	$\approx 2^{80}$
10	5	2	8	65521	4	4	180.8 s.	$\approx 2^{80}$
12	3	4	0	65521	1	1	0.1 s.	
12	3	4	5	65521	1	1	0.9 s.	$\approx 2^{96}$
12	3	4	6	65521	1	1	0.9 s.	$\approx 2^{96}$
12	3	4	7	65521	2	2	20.5 s.	$\approx 2^{96}$

12	3	4	8	65521	2	2	25.2 s.	$\approx 2^{96}$
12	3	4	9	65521	3	3	414 s.	$\approx 2^{96}$
20	5	4	0	65521	0	0	1.6 s.	
20	5	4	5	65521	1	1	55.2 s.	$\approx 2^{160}$
20	5	4	10	65521	1	1	78.9 s.	$\approx 2^{160}$
20	10	2	10	65521	1	1	78.8 s.	$\approx 2^{160}$
20	2	10	10	65521	1	1	78.7 s.	$\approx 2^{160}$
24	6	4	0	65521	0	0	4.9 s.	
24	6	4	12	65521	1	1	376.1 s.	$\approx 2^{192}$
30	15	2	15	65521	1	1	2910.5 s.	$\approx 2^{160}$
32	8	4	0	65521	0	0	31.3 s.	
32	8	4	10	65521	1	1	3287.9 s.	$\approx 2^{256}$
32	8	4	16	65521	1	1	4667.9 s.	$\approx 2^{256}$
36	18	2	15	65521	1	1	13427.4 s.	$\approx 2^{256}$

Interpretation of the results
Let us mention that $n = 16$ and $n = 32$ were two challenges proposed by the designers of $2\mathrm{R}^{-}$ schemes. First it should be observed that the parameters b and n_i of the S-box functions seem irrelevant for the complexity of our algorithm. We also tested our algorithm for instances of FDP constructed with various forms of ψ, ϕ (C^*+S-Box functions, Triangular+S-Box functions,...) and several values of q. These results are very similar to the ones obtained for S-Box functions, and thus not quoted here. The major observation is that our algorithm behaves exactly as predicted. That is, $d_{theo} = \lceil \frac{n}{u} - 1 \rceil$ is exactly equal to the d_{real} observed in practice.

Acknowledgements

We thank Lilian Bohy, Jintai Ding and anonymous referees for numerous comments which improved the presentation of the results.

References

1. A.W. Adams and P. Loustaunau. *An Introduction to Gröbner Bases.* Graduate Studies in Mathematics, Vol. 3, AMS, 1994.
2. G. Ars, J.-C. Faugère, H. Imai, M. Kawazoe, and M. Sugita. *Comparison Between XL and Gröbner Basis Algorithms.* Advances in Cryptology – ASIACRYPT 2004, Lecture Notes in Computer Science, vol. 3329, pp. 338-353, 2004.
3. G. Ars, and J.-C. Faugère. *Algebraic Immunities of functions over finite fields.* Proceedings of BFCA'05, Rouen, 2005. Also available at `http://eprint.iacr.org/2004/188.ps`.
4. M. Bardet, J-C. Faugère, B. Salvy and B-Y. Yang. *Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems.* In MEGA 2005, Eighth International Symposium on Effective Methods in Algebraic Geometry, 15 pages, 2005.
5. M. Bardet, J-C. Faugère, and B. Salvy. *On the Complexity of Gröbner Basis Computation of Semi-Regular Overdetermined Algebraic Equations.* In Proc. of International Conference on Polynomial System Solving (ICPSS), pp. 71–75, 2004.
6. E. Biham. *Cryptanalysis of Patarin's 2-Round Public Key System with S Boxes (2R).* Advances in Cryptology – CRYPTO 2000, Lecture Notes in Computer Science, vol. 1807, Springer–Verlag, pp. 408-416, 2000.
7. B. Buchberger. *Gröbner Bases : an Algorithmic Method in Polynomial Ideal Theory.* Recent trends in multidimensional systems theory. Reider ed. Bose, 1985.
8. B. Buchberger, G.-E. Collins, and R. Loos. *Computer Algebra Symbolic and Algebraic Computation.* Springer-Verlag, second edition, 1982.

9. V. Carlier, H. Chabanne, and E. Dottax *Grey Box Implementation of Block Ciphers Preserving the Confidentiality of their Design.* Proceedings of BFCA'05, Rouen, 2005. Also available at `http://eprint.iacr.org/2004/188.ps`.
10. N. Courtois, L. Goubin, and J. Patarin. *SFLASH, a Fast Asymmetric Signature Scheme for low-cost Smartcards – Primitive Specification and Supporting Documentation.* Available at `http://www.minrank.org/sflash-b-v2.pdf`.
11. 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.
12. M. Dickerson. *The functional Decomposition of Polynomials.* Ph.D Thesis, TR 89-1023, Departement of Computer Science, Cornell University, Ithaca, NY, July 1989.
13. J.-C. Faugère. *A New Efficient Algorithm for Computing Gröbner Casis without Reduction to Zero:* F_5. Proceedings of ISSAC, pp. 75–83. ACM press, July 2002.
14. J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. *Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering.* Journal of Symbolic Computation, 16(4), pp. 329–344, 1993.
15. D. Kozen, and S. Landau. *Polynomial decomposition algorithms.* J. Symb. Comput. (7), pp 445–456, 1989.
16. L. Goubin, and J. Patarin. *Trapdoor One-way Permutations and Multivariate Polynomials.* Information and Communication Security, First International Conference (ICICS'97), Lecture Notes in Computer Science vol. 1334, Springer–Verlag, pp. 356–368, 1997.
17. 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.
18. L. Goubin, and J. Patarin. *Asymmetric Cryptography with S-Boxes – Extended Version.* Available at `http://citeseer.ist.psu.edu/patarin97asymmetric.html`.
19. J. Gutierrez, R. Rubio, J. von zur Gathen. *Multivariate Polynomial Decomposition.* Algebra in Engineering, Communication and Computing, 14 (1), pp. 11–31.
20. T. Matsumoto, and H. Imai. *Algebraic Methods for Constructing Asymmetric Cryptosystems.* Algebraic and Error-Correcting Codes. Prod. Third Intern. Conf., Grenoble, France, Springer-Verlag, pp. 108–119, 1985.
21. T. Matsumoto, and H. Imai. *Public Quadratic Polynomial-tuples for efficient signature-verification and message-encryption.* Advances in Cryptology – EUROCRYPT 1988, Lecture Notes in Computer Science, vol. 330, Springer–Verlag, pp. 419–453, 1988.
22. J. Patarin. *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88.* Advances in Cryptology – CRYPTO 1995, Lecture Notes in Computer Science, Springer-Verlag, vol. 963, pp. 248-261, 1995.
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. 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.
26. 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.

Appendix A

The F_5 algorithm

The historical method for computing Gröbner bases is Buchberger's algorithm [8, 7]. Recently, more efficient algorithms have been proposed. To date, F_5 [13] is the most efficient for computing Gröbner bases. In a nutshell, this algorithm constructs incrementally the following matrices in degree d:

$$A_d = \begin{array}{c} \\ t_1 f_1 \\ t_2 f_2 \\ t_3 f_3 \\ \dots \end{array} \begin{array}{c} m_1 \succ m_2 \succ m_3 \dots \\ \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \end{array}$$

where the indices of the columns are monomials sorted for the admissible ordering $\prec$ and the rows are product of some polynomials f_i by some monomials t_j such that $\deg(t_j f_i) \leq d$. For a regular system [13] the matrices A_d are of full rank. In a second step, row echelon forms of theses matrices are computed, i.e.

$$A'_d = \begin{array}{c} \\ t_1 f_1 \\ t_2 f_2 \\ t_3 f_3 \\ \dots \end{array} \begin{array}{c} m_1 \ m_2 \ m_3 \dots \\ \begin{bmatrix} 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix} \end{array}$$

Note that for each d, A'_d contains a d-Gröbner basis of the ideal considered. Important parameters to evaluate the complexity of F_5 is the maximal degree d occurring in the computation and the size of the matrix A_d. The overall cost is thus dominated by $(\#A_d)^3$. Very roughly, $(\#A_d)$ can be approximated by $O(n^d)$. A more precise complexity analysis can be found in [4, 5].