Polynomial Equivalence Problems: Algorithmic and Theoretical Aspects

L. Perret, J.C. Faugere (Eurocrypt 2006)

November 21, 2005

Abstract

The Isomorphism of Polynomial (IP) [27], which is the main concern of this paper, originally corresponds to the problem of recovering the secret key of a C^* scheme [26]. Besides, the security of various other schemes (e.g. signature, authentication [27], traitor tracing [5], *etc...*) also depends on the practical hardness of IP. Due to its numerous applications, the Isomorphism of Polynomial is thus one of the most fundamental problems in multivariate cryptography. In this paper, we address two complementary aspects of IP, namely its theoretical and practical difficulty. We present an upper bound on the theoretical complexity of "IP-like" problems, i.e. a problem consisting in recovering a particular transformation between two sets of multivariate polynomials. Typical "IP-like" problems are for instance the Isomorphism of Polynomial with one Secret (IP1S) [27], and IP indeed. Our bound is obtained by introducing a generic description of these problems permitting to use classical results of groups theory. Concerning the practical aspect, we present a new algorithm for solving IP. In a nutshell, the idea is to generate a suitable algebraic system of equations whose zeroes correspond to a solution of IP. From a practical point of view, we employed a fast Gröbner bases algorithm, namely F_5 [17], for solving this system. This approach is efficient in practice and obliges to modify the current security criteria for IP. We have indeed broken several challenges proposed in the literature [27, 28, 5]. For instance, we solved a challenge proposed by O. Billet and H. Gilbert at Asiacryt'03 [5] in less than one second.

Keywords : Public-Key Cryptography, Cryptanalysis, Isomorphism of Polynomial (IP), Gröbner bases, F_5 algorithm

1 Introduction

Multivariate cryptography – which can be roughly defined as the cryptography using polynomials in several variables – offers a relatively wide spectrum of problems that can be used in public-key cryptography. The Isomorphism of Polynomial (IP) lies in this family [27]. Briefly, this problem consists in recovering a particular transformation between two sets of multivariate polynomials permitting to obtain one set from the other. It originally corresponds to the problem of recovering the secret key of a C^* scheme [26]. Besides, the security of several other schemes are directly based on the practical difficulty of IP, namely the authentication/signature schemes proposed by J. Patarin at Eurocrypt'96 [27], and the traitor tracing scheme described by O. Billet and H. Gilbert at Asiacrypt'03 [5]. We also mention that IP is in a certain manner related to the security of Sflash [13] – the signature scheme

recommended by the European consortium Nessie for low-cost smart cards[1] – and can be alternatively viewed as the problem of detecting affine equivalence between S-Boxes [6].
All in all, one can consider the hardness of IP as one of the major issues in multivariate cryptography. The goal of this paper is to provide new insights on the theoretical and practical complexity of IP and some its relevant variants.

1.1 Previous Works

To the best of our knowledge, the most significant results concerning IP are presented in [11]. This paper gives an upper bound on the theoretical complexity of IP. Nevertheless, we point out that the proof provided is actually not complete[2]. Anyway, the upper bound presented in this paper is original and general. It is indeed based on a group theoretical approach of IP and actually dedicated to "IP-like" problems. A new algorithm for solving IP, called "To and Fro", is also described in [11]. This algorithm is however devoted to special instances[3] of IP, namely the ones corresponding to a public key of C^* [26]. Thus, it can not be used for solving generic instances of IP, contrarily to the algorithm presented here. Besides, we present in section 4 experimental results demonstrating that our algorithm outperforms the "To and Fro" method. Finally, we mention a result due to W. Geiselmann, R. Steinwandt, and T. Beth [23]. In the context of C^*, they showed how to easily recover the affine parts of a solution of IP[4]. A similar property also holds in the context of HFE [20]. Such a kind of result does not exists for generic instances of IP. It means nevertheless that – in the cryptographic context – we can focus our attention to the linear variant of IP, named 2PLE here.

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 of our algorithm, namely varieties and Gröbner bases. Recent algorithms (i.e. F_4 and F_5 [16, 17]) computing these bases are also succinctly described. Finally, we define more formally the Isomorphism of Polynomial (IP) and two of its variants, namely the Isomorphism of Polynomial with one Secret (IP1S) [27], and the linear variant of IP that we named 2PLE. In section 3, we show that these problems are actually particular instances of a more general problem that we called Polynomial Equivalence (PE). This problem provides a formal definition of an "IP-like" problem. Using a classical results of groups theory, we conclude this section by providing an upper bound on the theoretical hardness of PE. A new algorithm for solving 2PLE is presented in section 4. In a nutshell, the idea is to generate a suitable polynomial system of equations whose zeroes correspond to a solution of IP. In order to construct this system, we also provide some specific properties of 2PLE. From a practical point of view, we used the most recent (and efficient) Gröbner bases algorithm, namely F_5 [17], for solving this system. It is really difficult to obtain a complexity bound really reflecting the practical behaviour of the F_5 algorithm. We therefore carried out experimental results illustrating the practical efficiency of our approach. We have indeed broken several challenges proposed the literature [27, 28, 5]. Precisely, we solved a challenge proposed by O. Billet and H. Gilbert at Asiacryt'03 [5] in less than one second. We also present experimental

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

[2] Precisely, the completeness and soundness properties of the interactive protocol have not been proved.

[3] Note that A. Biryukov, C. De Cannière *et al* proposed an adaptation of this algorithm for detecting affine equivalence between S-Boxes [6].

[4] This results is in fact proposed in the context of Sflash [13], but can be easily extended to C^*.

evidences that our algorithm solves a large classes of instances of IP (used in [27, 28]) in polynomial-time.

2 Preliminaries

The notations used throughout this paper are the following. We denote by $\mathbb{F}_q$, the finite field with $q = p^r$ elements (p a prime, and $r \geq 1$), by $\mathcal{M}_{n,u}(\mathbb{F}_q)$ the set of $n \times u$ matrices whose components are in $\mathbb{F}_q$. As usual, $GL_n(\mathbb{F}_q)$ represents the set of invertible matrices of $\mathcal{M}_{n,n}(\mathbb{F}_q)$, and $AGL_n(\mathbb{F}_q)$ denotes the cartesian product $GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$. Lastly, $\underline{x} = (x_1, \ldots, x_n)$, $\mathbb{F}_q[\underline{x}] = \mathbb{F}_q[x_1, \ldots, x_n]$ the polynomial ring in the n indeterminate $x_1, \ldots, x_n$ over $\mathbb{F}_q$, and $p(\underline{x})$ stands for $p(x_1, \ldots, x_n)$. By convention, a underlined letter will always refer to a vector.

2.1 Gröbner bases

We define now two essential notions of this paper, namely varieties and Gröbner bases. For a more thorough introductions to these tools, we refer to [1, 15]. Let then $\underline{p} = (p_1, \ldots, p_s)$ be polynomials of $\mathbb{F}_q[\underline{x}]$. We shall say that a vector $\underline{z} \in \mathbb{F}_q^n$ is a *zero* of $\underline{p}$, if $p_i(\underline{z}) = 0$, for all $i, 1 \leq i \leq s$, and we shall call $\mathcal{I} = \langle p_1, \ldots, p_s \rangle = \left\{ \sum_{k=1}^{s} p_k u_k, u_1, \ldots, u_k \in \mathbb{F}_q[\underline{x}] \right\} \subset \mathbb{F}_q[\underline{x}]$ the *ideal generated* by $p_1, \ldots, p_s$. We shall also denote by $V(\mathcal{I}) = \{\underline{z} \in \mathbb{F}_q^n : p(\underline{z}) = 0, \forall p \in \mathcal{I}\}$ the *variety associated* to $\mathcal{I}$. Since every element of $\mathcal{I}$ vanishes at each zero of $\underline{p}$, we have equivalently $V(\mathcal{I}) = \{\underline{z} \in \mathbb{F}_q^n : p_i(\underline{z}) = 0, \forall i, 1 \leq i \leq s\}$. Gröbner bases provide a method for computing this object. Informally, a Gröbner basis of an ideal $\mathcal{I}$ is a computable generator set of $\mathcal{I}$ with "good" algorithmic properties. It is defined with respect to a *monomial ordering.* For instance, the *Lexicographic order* (Lex):

$$x_1^{\alpha_1} \cdots x_n^{\alpha_n} <_{Lex} x_1^{\beta_1} \cdots x_n^{\beta_n} \iff \exists\, i_0 \leq n, \text{ such that } \alpha_{i_0} < \beta_{i_0} \text{ and } \alpha_i = \beta_i, \forall i = 1 \ldots i_0 - 1,$$

is a classical monomial ordering. To define Gröbner bases, we need the following definitions.

Definition 2.1. *For any n-uple $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, we denote by $\mathbf{x}^\alpha$ the* monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. *We shall define the* total degree *of this monomial by the sum $\sum_{i=1}^{n} \alpha_i$. The* leading monomial *of a polynomial $p \in \mathbb{F}_q[\underline{x}]$ is the largest monomial – w.r.t some monomial ordering $<$ – among the monomials of p. This leading monomial will be denoted by $LM(p, <)$. The* degree *of p – noted deg(p) – is the total degree of $LM(p, <)$. Finally, the* maximal total degree *of p is the maximal total degree of the monomials occurring in p.*

We are thus in a position to define one of the main object of this paper.

Definition 2.2. *A set of polynomials G is a* Gröbner basis *– w.r.t. a monomial ordering $<$ – of an ideal $\mathcal{I} \subset \mathbb{F}_q[\underline{x}]$, if for all $p \in \mathcal{I}$ there exists $g \in G$ such that $LM(g, <)$ divides $LM(p, <)$.*

Gröebner bases are a fundamental tool to study algebraic systems in both theory and practice, providing an algorithmic solution to several problems related to polynomial systems[5]. For example, the one of deciding the membership of a polynomial to a given ideal. Anyway, we pay here a particular attention to Gröbner bases computed for elimination ordering. They indeed offer a way of simplifying an algebraic system by procuring an equivalent system with

[5] Some of the problems can de found in [1] for instance.

a structured shape. A lexicographical Gröbner basis of a zero-dimensional system (i.e. with a finite number of zeroes over the algebraic closure[6]) is for instance always as follows:

$$\left\{f(x_1) = 0, f_2(x_1,x_2) = 0, \ldots, f_{k_2}(x_1,x_2) = 0, \ldots, f_{k_{n-1}+1}(\underline{x}) = 0, \cdots, f_{k_n}(\underline{x}) = 0\right\}.$$

To compute the variety, we simply have to successively eliminate variables by computing zeroes of univariate polynomials and back-substituting results. Hence, the variety associated to a zero dimensional system $p_1 = 0, \ldots, p_s = 0$ can be easily deduced from a lexicographical Gröbner basis of $\mathcal{I} = \langle p_1, \ldots, p_s \rangle$. Computing a Gröbner basis w.r.t. a lexicographical order is however in practice much slower than computing a Gröbner basis w.r.t. another monomial ordering[7] of the same ideal. Algorithms changing the monomial ordering of a Gröbner basis permit to handle this problem. The FLGM algorithm [19] for instance allows – in the zero-dimensional case – to transform a Gröbner basis w.r.t. some monomial ordering into a lexicographical Gröbner basis in polynomial-time.

The historical method for computing Gröbner bases is Buchberger's algorithm [9, 8]. It has several variants and it is implemented in most of general computer algebra systems like Maple or Mathematica. Recently, more efficient algorithms have been proposed to compute Gröbner bases. The F_4 algorithm [16] is based on the intensive use of linear algebra methods. In short, the arbitrary choices – which limit the practical efficiency of Buchberger's algorithm – are left to computational strategies related to classical linear algebra problems (mainly the computation of a row echelon form). In [17], a new criterion (the F_5 criterion) for detecting useless computations has been provided. It has to be mentioned that Buchberger's algorithm spends 90% of its time to perform these useless computations. Under some regularity conditions on the system, it has been proved that all useless computations can be avoided. A new algorithm, named F_5, has been then built using this criterion and linear algebra methods. In a nutshell, it constructs incrementally the following matrices in degree d:

$$A_d = \begin{array}{c} \\ t_1 f_1 \\ t_2 f_2 \\ t_3 f_3 \\ \cdots \end{array} \begin{array}{c} \begin{array}{cccc} m_1 > & m_2 > & m_3 & \ldots \end{array} \\ \begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \end{array}$$

where the indices of the columns are monomials sorted for the admissible ordering $<$ 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 the matrices A_d are full rank. In a second step, row echelon forms of the matrices are computed, i.e.

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

[6]In the cryptographic setting, the systems considered are almost all the times zero-dimensional. The field equations can be indeed added to the systems.

[7]It is usually for the Degree Reverse Lexicographical order (see [1] for a definition) that the computation of Gröbner bases are in practice the fastest.

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 $(\#A_d)^3$. A more precise complexity analysis can be found in [3, 4].
Even if F_5 still computes the same mathematical object (a Gröbner basis), the gap with existing other algorithms is consequent. The F_5 algorithm is then the best to date for computing Gröbner bases, and hence zero-dimensional varieties. In particular, it has been proved [2] – from both a theoretic and practical point of view – that XL [14] is less efficient than F_5. Recall that XL is a method, proposed at Eurocrypt'2000 [14], permitting to recover the solution of a zero-dimensional system. Anyway, due to the range of examples that become computable with F_5, Gröbner basis can be considered as a reasonable computable object in real scale applications. For systems arising in cryptography, F_5 has for instance given impressing results on the HFE challenge [18]. We present in this paper a new application of F_5 in cryptography.

2.2 Isomorphism of Polynomial and Related Problems

Before defining formally IP, we briefly come back in this part to the origin of this problem. Let's then describe the encryption scheme called C* [26]. The public key of this system is a set of multivariate quadratic polynomials $\underline{b} = \big(b_1(\underline{x}), \ldots, b_n(\underline{x})\big) \in \mathbb{F}_q[\underline{x}]^n$. These polynomials are obtained by applying two bijective affine transformations – represented by the pairs $(S, \underline{T})$ and $(U, \underline{V})$ of $AGL_n(\mathbb{F}_q)$ – to particular polynomials $\underline{a} = \big(a_1(\underline{x}), \ldots, a_n(\underline{x})\big) \in \mathbb{F}_q[\underline{x}]^n$, i.e.:

$$\big(b_1(\underline{x}), \ldots, b_n(\underline{x})\big) = \big(a_1(\underline{x}S + \underline{T}), \ldots, a_n(\underline{x}S + \underline{T})\big)U + \underline{V}.$$

we shall denote this equality by $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S + \underline{T})U + \underline{V}$ in the sequel.
The encryption process simply consists in evaluating a message $\underline{m} \in \mathbb{F}_q^n$ on $\underline{b}$, i.e.:

$$\underline{c} = \big(b_1(\underline{m}), \ldots, b_n(\underline{m})\big).$$

To recover the correct plaintext, the legitimate recipient uses the invertibility of the affine transformations combined with the particular structure of the polynomials of $\underline{a}$. These polynomials are actually the "multivariate representation" of a univariate monomial over a degree n extension of $\mathbb{F}_q$. Finding a root of this univariate polynomial is then equivalent to recovering a zero of $\underline{a}$. For practical reasons this univariate monomial is specially chosen to induce a bijection. Due to this constraint, the polynomials of $\underline{a}$ are always considered as a public data. The secret key of C* is then constituted of the two pairs $(S, \underline{T})$ and $(U, \underline{V})$ of $AGL_n(\mathbb{F}_q)$.

The first approach for attacking this scheme consists in trying to retrieve the message corresponding to a ciphertext $\underline{c} \in \mathbb{F}_q^n$, i.e. finding a zero of $\underline{b}(\underline{x}) = \underline{c}$. That is solving an instance of the so-called MQ problem, which has been proved NP-Hard [10, 22]. We emphasize nevertheless that such a kind of result uniquely guarantee the worst-case hardness and does not provide any information concerning the average-case difficulty. For instance, J.-C. Faugère and A. Joux proposed a polynomial-time algorithm for solving instances of MQ corresponding to the public key of HFE [18], which is an extension of C*.

Another approach for breaking C* consists in attempting to recover the affine transformations hiding the structure of $\underline{a}$, i.e. extracting the secret key from the public key. This problem, introduced by J. Patarin at Eurocrypt'96 [27], is defined as follows:
Isomorphism of Polynomial (IP)

Input: $\underline{a} = (a_1, \ldots, a_u)$, and $\underline{b} = (b_1, \ldots, b_u)$ in $\mathbb{F}_q[\underline{x}]^u$.
Question: Find – if any – $(S, \underline{T}) \in AGL_n(\mathbb{F}_q)$ and $(U, \underline{V}) \in AGL_u(\mathbb{F}_q)$, such that:

$$\underline{b}(\underline{x}) = \underline{a}(\underline{x}S + \underline{T})U + \underline{V}.$$

It is however without solving any of the two problems mentioned above that J. Patarin proposed a complete cryptanalysis of C* [29]. This attack uses the very particular structure of the polynomials of $\underline{a}$. This result does not then affect at all the practical hardness of IP.

In the practical applications, it is more precisely the linear variant of IP which is usually considered [27, 5], i.e. when the vectors $\underline{T}$ and $\underline{V}$ are both equal to the null vector. This problem, that we named 2PLE, is the following:
Input: $\underline{a} = (a_1, \ldots, a_u)$, and $\underline{b} = (b_1, \ldots, b_u)$ in $\mathbb{F}_q[\underline{x}]^u$.
Question: Find – if any – $(S, U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$, such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)U$.

The security estimate provided for this problem [28] is based on the complexity of the "To and Fro" (TF) algorithm [11, 12], which is $q^{n/2}$ for quadratic polynomials, and q^n otherwise.

Remark 2.1. *In the rest of this paper, we will always suppose that all the polynomials of $\underline{a}$ have the same maximal total degree noted D. In the practical applications, we have $2 \leq D \leq 4$. Note that, if $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)U$, for some $(S, U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$, then the polynomials of $\underline{b}$ must have the same maximal total degree that the ones of $\underline{a}$, i.e. D.*

3 Polynomial Equivalence Problems: A Unified Point of View

The Isomorphism of Polynomial, and 2PLE problems have actually a very similar formulation. An input of these problems is formed of two systems of multivariate polynomials and the question consists in recovering a particular transformation permitting to express one system in function of the other. All transformations have the same characteristic: inducing a group action on $\mathbb{F}_q[\underline{x}]^u$. Recall that for a group $(G, \cdot)$ with identity element e_G, we shall say that a map $\phi : G \times \mathbb{F}_q[\underline{x}]^u \to \mathbb{F}_q[\underline{x}]^u$ *acts* on $\mathbb{F}_q[\underline{x}]^u$, if we have the two following conditions:

$$\begin{cases} \phi(e_G, \underline{p}) = \underline{p}, \text{ for all } \underline{p} = (p_1, \ldots, p_u) \in \mathbb{F}_q[\underline{x}]^u, \\ \phi\big(g, \phi(g', \underline{p})\big) = \phi(g \cdot g', \underline{p}), \text{ for all } g, g' \in G, \text{ and for all } \underline{p} = (p_1, \ldots, p_u) \in \mathbb{F}_q[\underline{x}]^u. \end{cases}$$

For 2PLE, one can then easily check that[8] $GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$ acts on $\mathbb{F}_q[\underline{x}]^u$ through:

$$\begin{array}{rccc} \phi_{\text{2PLE}}: & GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q) \times \mathbb{F}_q[\underline{x}]^u & \to & \mathbb{F}_q[\underline{x}]^u \\ & \big((S, U), \underline{a}\big) & \mapsto & \underline{a}(\underline{x}S)U \end{array}$$

Similarly for IP, $AGL_n(\mathbb{F}_q) \times AGL_u(\mathbb{F}_q)$ acts on $\mathbb{F}_q[\underline{x}]^u$ through:

$$\begin{array}{rccc} \phi_{\text{IP}}: & AGL_n(\mathbb{F}_q) \times AGL_u(\mathbb{F}_q) \times \mathbb{F}_q[\underline{x}]^u & \to & \mathbb{F}_q[\underline{x}]^u \\ & \big((S, \underline{T}), (U, \underline{V}), \underline{a}\big) & \mapsto & \underline{a}(\underline{x}S + \underline{T})U + \underline{V} \end{array}$$

This observation naturally leads to the introduction of the following problem.
Let $(G, \cdot)$ be a group, and $\phi : G \times \mathbb{F}_q[\underline{x}]^u \to \mathbb{F}_q[\underline{x}]^u$ be an action of G on $\mathbb{F}_q[\underline{x}]^u$.

[8] In order to simplify the notations, we identify here a group $(G, \cdot)$ with its set G.

Given $(\underline{a}, \underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$, the problem that we named *Polynomial Equivalence*, with respect to $(G, \cdot)$ and ϕ – noted $\mathrm{EP}\big(G, \phi\big)$ – is then the one of finding if any $g \in G$, verifying:

$$\underline{b} = \phi(g, \underline{a}),$$

denoted $\underline{a} \equiv_{(G,\phi)} \underline{b}$ in the sequel. This problem is very convenient since it procures a unified description of IP and 2PLE. Indeed, $\mathrm{PE}\big(GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q), \phi_{\mathrm{2PLE}}\big)$=2PLE, and $\mathrm{PE}\big(AGL_n(\mathbb{F}_q) \times AGL_u(\mathbb{F}_q), \phi_{\mathrm{IP}}\big)$=IP. More generally, PE provides a unified description of "IP-like" problems. In our mind, such a kind of problem consists in recovering a particular transformation between two sets of multivariate polynomials. For instance, the Isomorphism of Polynomial with one Secret (IP1S) – introduced at Eurocrypt'96 by J. Patarin [27] – falls in this new formalism. This problem, which can be used to design an authentication (resp. signature) scheme [27], is as follows: Given $(\underline{a}, \underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$, IP1S is the problem of finding – if any – $(S, \underline{T}) \in AGL_n(\mathbb{F}_q)$, such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S + \underline{T})$. Using our formalism, we immediately obtain that $\mathrm{PE}\big(AGL_n(\mathbb{F}_q), \phi_{\mathrm{IP1S}}\big)$ = IP1S, with $\phi_{\mathrm{IP1S}} : AGL_n(\mathbb{F}_q) \times \mathbb{F}_q[\underline{x}]^u \rightarrow \mathbb{F}_q[\underline{x}]^u$, $\big((S, \underline{T}), \underline{a}(\underline{x})\big) \mapsto \underline{a}(\underline{x}S + \underline{T})$. Finally, the following lemma justifies the use of the word equivalence in PE.

Lemma 3.1. *Let $(G, \cdot)$ be a group, and $\phi : G \times \mathbb{F}_q[\underline{x}]^u \rightarrow \mathbb{F}_q[\underline{x}]^u$ be an action of G on $\mathbb{F}_q[\underline{x}]^u$. $\equiv_{(G,\phi)}$ is an equivalence relation on $\mathbb{F}_q[\underline{x}]^u$.*

Proof. Let $\underline{a}, \underline{b}, \underline{c} \in \mathbb{F}_q[\underline{x}]^u$, and e_G be the identity element of G. First, we have $\underline{a} = \phi(e_G, \underline{a})$, i.e. $\underline{a} \equiv_{(G,\phi)} \underline{a}$. Now, if $\underline{b} = \phi(g, \underline{a})$, for some $g \in G$, then $\phi(g^{-1}, \underline{b}) = \phi\big(g^{-1}, \phi(g, \underline{a})\big) = \phi(g^{-1} \cdot g, \underline{a}) = \underline{a}$. That is, $\underline{a} \equiv_{(G,\phi)} \underline{b} \Longrightarrow \underline{b} \equiv_{(G,\phi)} \underline{a}$. Finally, let $g, g' \in G$ such that $\underline{b} = \phi(g, \underline{a})$ and $\underline{c} = \phi(g', \underline{b})$. Thus, $\underline{c} = \phi\big(g', \phi(g, \underline{a})\big) = \phi\big(g' \cdot g, \underline{a}\big)$, i.e. $\underline{a} \equiv_{(G,\phi)} \underline{b}$ and $\underline{b} \equiv_{(G,\phi)} \underline{c}$ implies that $\underline{a} \equiv_{(G,\phi)} \underline{c}$. □

3.1 Polynomial Equivalence Problems and Group theory

In the Graph Isomorphism context, the introduction of groups theory concepts permitted to achieve significant advances from both a theoretical and algorithmic point of view [25, 21]. The formalism previously inserted permits to naturally extend these results to Polynomial Equivalence problems. Let's then introduce the following definitions.

Definition 3.1. *Let $(G, \cdot)$ be a group, and $(\underline{a}, \underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$. We shall call $Aut_{(G,\phi)}(\underline{a}) = \big\{g \in G : \phi(g, \underline{a}) = \underline{a}\big\}$, and $Aut_{(G,\phi)}(\underline{b}) = \big\{g \in G : \phi(g, \underline{b}) = \underline{b}\big\}$, the* automorphism groups of $\underline{a}$ and $\underline{b}$ with respect to (G, ϕ). *We shall also set $S_{(G,\phi)}(\underline{a}, \underline{b}) = \big\{g \in G : \underline{b} = \phi(g, \underline{a})\big\}$.*

Remark 3.1. *$Aut_{(G,\phi)}(\underline{a})$ $\big($resp. $Aut_{(G,\phi)}(\underline{b})\big)$ is also known as* stabilizer *of $\underline{a}$ (resp. $\underline{b}$) with respect to (G, ϕ). However, we will rather call these sets automorphism groups. This designation being indeed more usually used in the Graph Isomorphism context [25]. The results that we are going to expose are then classical results of groups theory concerning the stabilizers and orbits. The proofs are nevertheless provided for the sake of completeness.*

We exhibit relations linking the automorphism groups of $\underline{a}$, $\underline{b}$ and $S_{(G,\phi)}(\underline{a}, \underline{b})$.

Proposition 3.1. *Let $(G, \cdot)$ be a group, and $\phi : G \times \mathbb{F}_q[\underline{x}]^u \rightarrow \mathbb{F}_q[\underline{x}]^u$ be an action of G on $\mathbb{F}_q[\underline{x}]^u$. Let also $(\underline{a}, \underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$. If there exists $g \in G$, such that $\underline{b} = \phi(g, \underline{a})$, then*

$S_{(G,\phi)}(\underline{a},\underline{b})$ *is a left (resp. right) coset – in* G *– of the automorphism group* $Aut_{(G,\phi)}(\underline{a})$ *(resp.* $Aut_{(G,\phi)}(\underline{b})$*). In other words:*

$$\begin{cases} S_{(G,\phi)}(\underline{a},\underline{b}) = \{g \cdot h : h \in Aut_{(G,\phi)}(\underline{a})\} = g \cdot Aut_{(G,\phi)}(\underline{a}), \\ S_{(G,\phi)}(\underline{a},\underline{b}) = \{h \cdot g : h \in Aut_{(G,\phi)}(\underline{b})\} = Aut_{(G,\phi)}(\underline{b}) \cdot g. \end{cases}$$

Proof. Let's show $S_{(G,\phi)}(\underline{a},\underline{b}) = g \cdot Aut_{(G,\phi)}(\underline{a})$.
First, take $g' \in S_{(G,\phi)}(\underline{a},\underline{b})$, i.e. $\underline{b} = \phi(g', \underline{a})$. We then have $\phi(g', \underline{a}) = \phi(g, \underline{a})$, and thus:

$$\phi(g^{-1} \cdot g', \underline{a}) = \phi\big(g^{-1}, \phi(g', \underline{a})\big) = \phi\big(g^{-1}, \phi(g, \underline{a})\big) = \phi(g^{-1} \cdot g, \underline{a}) = \underline{a}.$$

Therefore, $g^{-1} \cdot g' \in Aut_{(G,\phi)}(\underline{a})$, i.e. $g' \in g \cdot Aut_{(G,\phi)}(\underline{b})$.
Now, let $q \in g \cdot Aut_{(G,\phi)}(\underline{a})$, i.e. $q = g \cdot g'$, with $g' \in Aut_{(G,\phi)}(\underline{a})$. By hypothesis:

$$\phi(g^{-1}, \underline{b}) = \phi\big(g^{-1}, \phi(g, \underline{a})\big) = \underline{a} = \phi(g', \underline{a}).$$

Therefore $\underline{b} = \phi\big(g, \phi(g^{-1}, \underline{b})\big) = \phi\big(g, \phi(g', \underline{a})\big) = \phi(g \cdot g', \underline{a})$, i.e. $q = g \cdot g' \in S_{(G,\phi)}(\underline{a},\underline{b})$.
The equality $S_{(G,\phi)}(\underline{a},\underline{b}) = Aut_{(G,\phi)}(\underline{a}) \cdot g$ can be proved in a similar way. $\square$

Consequently:

Corollary 3.1. *Let* $(G, \cdot)$ *be a group, and* $\phi : G \times \mathbb{F}_q[\underline{x}]^u \to \mathbb{F}_q[\underline{x}]^u$ *be an action of* G *on* $\mathbb{F}_q[\underline{x}]^u$*. Let also* $(\underline{a},\underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$*. If* $\underline{b} = \phi(g, \underline{a})$*, for some* $g \in G$*, then the automorphism groups* $Aut_{(G,\phi)}(\underline{a})$ *and* $Aut_{(G,\phi)}(\underline{b})$ *are conjugate, i.e.* $Aut_{(G,\phi)}(\underline{b}) = g \cdot Aut_{(G,\phi)}(\underline{a}) \cdot g^{-1}$*. Moreover:*

$$|S_{(G,\phi)}(\underline{a},\underline{b})| = |Aut_{(G,\phi)}(\underline{b})| = |Aut_{(G,\phi)}(\underline{a})|.$$

Proof. According to proposition 3.1, $g \cdot Aut_{(G,\phi)}(\underline{a}) = S_{(G,\phi)}(\underline{a},\underline{b}) = Aut_G(\underline{b}) \cdot g$. Thus, $Aut_{(G,\phi)}(\underline{b}) = g \cdot Aut_{(G,\phi)}(\underline{a}) \cdot g^{-1}$. The equality $|S_{(G,\phi)}(\underline{a},\underline{b})| = |Aut_{(G,\phi)}(\underline{b})| = |Aut_{(G,\phi)}(\underline{a})|$ is simply obtained by using proposition 3.1 and the first assertion of corollary 3.1. $\square$

3.2 A Generic Upper Bound on the Complexity of "IP-like" Problems

Using the Polynomial Equivalence problem previously defined, we give in this part a general upper bound on the theoretical complexity of "IP-like" problems. Let's then fix a group $(G, \cdot)$ acting on $\mathbb{F}_q[\underline{x}]^u$ through a map noted ϕ.

Remark 3.2. *For the sack of convenience, we suppose here that* G *is included in a finite set* $\mathcal{E}$*. We also suppose that the uniform distribution of the elements of* $\mathcal{E}$ *can be simulated in polynomial-time. These assumptions allows to facilitate the proofs, and are additionally well adapted to "IP-like" problems. Indeed,* $GL_n(\mathbb{F}_q) \subset \mathcal{M}_{n,n}(\mathbb{F}_q)$*,* $GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q) \subset \mathcal{M}_{n,n}(\mathbb{F}_q) \times \mathbb{F}_q^n$*,* $AGL_n(\mathbb{F}_q) \times AGL_u(\mathbb{F}_q) \subset \mathcal{M}_{n,n}(\mathbb{F}_q) \times \mathbb{F}_q^n \times \mathcal{M}_{u,u}(\mathbb{F}_q) \times \mathbb{F}_q^u$*.*

In order to obtain our upper bound, we introduce the following definition.

Definition 3.2. *An interactive proof for a language* L *(that is w.l.o.g. a subset of* $\{0,1\}^*$*) is a two party protocol between a verifier* $\mathcal{V}$ *and a prover* $\mathcal{P}$*. At the end of the protocol, the verifier has to accept or reject a given input such that the following conditions hold:*
Efficiency. *The verifier strategy is a probabilistic polynomial time procedure. (Note that the prover is computationally unbounded.)*
Completeness. *For all* $x \in L$*,* $Pr[(\mathcal{V}, \mathcal{P})(x) \text{ accepts}] = 1$*.*
Soundness. *For all* $x \notin L$*, and for any prover* $\mathcal{P}^*$*,* $Pr[(\mathcal{V}, \mathcal{P}^*)(x) \text{ accepts}] \leq \frac{1}{2}$*.*
The probabilities are taken over the random choices of the verifier.

Let's analyse the following two party protocol:

Input: $(\underline{a_0}, \underline{a_1}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$
Protocol: $\mathrm{PI}(G, \phi)$
The verifier chooses uniformly at random $i \in \{0, 1\}$.
He also chooses uniformly at random $g \in \mathcal{E}$ and checks if $g \in G$. If after C trials the verifier does not obtain an element $g \in G$, he accepts directly.
Otherwise, he sends $\underline{a'} = \phi(g, \underline{a_i})$ to the prover.
The prover replies by sending $j \in \{0, 1\}$ to the verifier.
The verifier accepts if $i = j$ and rejects otherwise.

Efficiency. The efficiency of this protocol depends on the cost of computing $\phi(g, \underline{a_i})$, for all $g \in G$, and of the number of trials C.

Completeness. If $\underline{a_0} \not\equiv_{(G,\phi)} \underline{a_1}$, then a prover can always check if $\underline{a'} \equiv_{(G,\phi)} \underline{a_0}$ or $\underline{a'} \equiv_{(G,\phi)} \underline{a_1}$. In this situation, the verifier accepts with probability one.

Soundness. If $\underline{a_0} \equiv_{(G,\phi)} \underline{a_1}$, then by transitivity $\underline{a'} \equiv_{(G,\phi)} \underline{a_1}$ and $\underline{a'} \equiv_{(G,\phi)} \underline{a_0}$. In such a case, we will show that $\underline{a'} = \phi(g, \underline{a_i})$ yields no information about the bit i chosen by the prover. Let then ψ be a random variable uniformly distributed over $\{0, 1\}$, and Σ be a random variable uniformly distributed over G.

Lemma 3.2. *Let $\underline{a_0}, \underline{a_1}, \underline{a'}$ be elements of $\mathbb{F}_q[\underline{x}]^u$. If $\underline{a_0} \equiv_{(G,\phi)} \underline{a_1}$ and $\underline{a'} \equiv_{(G,\phi)} \underline{a_0}$, then:*

$$Pr[\psi = 0 \,|\, \underline{a_\psi}(\underline{x}\Sigma) = \underline{a'}] = Pr[\psi = 1 \,|\, \underline{a_\psi}(\underline{x}\Sigma) = \underline{a'}] = \frac{1}{2}.$$

Proof. We have $\Pr[\phi(\Sigma, \underline{a_\psi}) = \underline{a'} \,|\, \psi = 0] = \Pr[\phi(\Sigma, \underline{a_0}) = \underline{a'}\,] = \Pr[\Sigma \in S_{(G,\phi)}(\underline{a_0}, \underline{a'})\,]$. Moreover, according to corollary 3.1:

$$|S_{(G,\phi)}(\underline{a_0}, \underline{a'})| = |Aut_{(G,\phi)}(\underline{a'})| = |S_{(G,\phi)}(\underline{a_1}, \underline{a'})|.$$

Therefore, $\Pr[\phi(\Sigma, \underline{a_0}) = \underline{a'}\,] = \Pr[\underline{a_1}(\underline{x}\Sigma) = \underline{a'}\,]$, and thus:

$$\Pr[\phi(\Sigma, \underline{a_\psi}) = \underline{a'} \,|\, \psi = 0] = \Pr[\phi(\Sigma, \underline{a_\psi}) = \underline{a'} \,|\, \psi = 1].$$

According to the Bayes formula:

$$\begin{aligned} \Pr \psi = 0 \,|\, \phi(\Sigma, \underline{a_\psi}) = \underline{a'}] &= \frac{\Pr[\psi{=}0]\Pr[\phi(\Sigma,\underline{a_\psi}){=}\underline{a'} \,|\, \psi{=}0]}{\Pr[\phi(\Sigma,\underline{a_\psi}){=}\underline{a'}]} \\ &= \frac{\Pr[\psi{=}1]\ \Pr[\phi(\Sigma,\underline{a_\psi}){=}\underline{a'} \,|\, \psi{=}1]}{\Pr[\phi(\Sigma,\underline{a_\psi}){=}\underline{a'}]} \\ &= \Pr[\psi = 1 \,|\, \phi(\Sigma, \underline{a_\psi}) = \underline{a'}]. \end{aligned}$$

Moreover:

$$\begin{aligned} \Pr[\psi = 0 \,|\, \phi(\Sigma, \underline{a_\psi}) = \underline{a'}] &= \frac{\Pr[\psi{=}0]\Pr[\phi(\Sigma,\underline{a_\psi}){=}\underline{a'} \,|\, \psi{=}0]}{\Pr[\phi(\Sigma,\underline{a_\psi}){=}\underline{a'}]} \\ &= \frac{\Pr[\psi{=}1]\ \Pr[\phi(\Sigma,\underline{a_0}){=}\underline{a'}]}{\Pr[\phi(\Sigma,\underline{a_\psi}){=}\underline{a'}]} \\ &= \frac{\Pr[\psi{=}1]\ \Pr[\Sigma{\in}S_{(G,\psi)}(\underline{a'},\underline{a_0})\,]}{\Pr[\Sigma{\in}S_{(G,\phi)}(\underline{a_\psi},\underline{a'})]} = \frac{1}{2} \end{aligned}$$

$\square$

Let then $\Pr[g \notin G | g \in \mathcal{E}]^C$ be the probability of not obtaining an element of G after C trials. From this lemma, it follows that no prover – no matter what its strategy is – can guess i with probability greater than $\frac{1}{2} + \Pr[g \notin G \,|\, g \in \mathcal{E}]^C$. Finally, using a classical result of R. B. Boppana, J. Hastad, and S. Zachos [7], we get that:

Corollary 3.2. *If the polynomial hierarchy not collapses then* IP*,* 2PLE*, and* IP1S *are not* NP-Hard*.*

Proof. Firstly, for all $g \in AGL_n(\mathbb{F}_q) \times AGL_u(\mathbb{F}_q)$ $\big($resp. $GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$, $AGL_n(\mathbb{F}_q)\big)$, one can compute $\phi_{\mathrm{IP}}(g, \underline{a'})$ $\big($resp. $\phi_{\mathrm{2PLE}}(g, \underline{a'}), \phi_{\mathrm{IP1S}}(g, \underline{a'})\big)$ in polynomial-time. Hence, the strategy of a verifier in $\mathrm{PI}\big(AGL_n(\mathbb{F}_q) \times AGL_u(\mathbb{F}_q), \phi_{\mathrm{IP}}\big)$, $\mathrm{PI}\big(GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q), \phi_{\mathrm{2PLE}}\big)$, and $\mathrm{PI}\big(AGL_n(\mathbb{F}_q), \phi_{\mathrm{IP1S}}\big)$ is efficient.
Let's give the number of trials in these protocols. We define L_{IP} as the language associated to IP, i.e. the set of instances of IP admitting a solution We also recall that more than $1/4$ of the matrices of $\mathcal{M}_{n,n}(\mathbb{F}_q)$ are invertible. Therefore for IP1S, we have $G = AGL_n(\mathbb{F}_q)$, $\mathcal{E} = \mathcal{M}_{n,n}(\mathbb{F}_q) \times \mathbb{F}_q^n$, and $\Pr[g \in G \,|\, g \in \mathcal{E}] \geq \frac{1}{4}$. By setting $C = 10$, we get that no prover can guess i with probability greater than $\frac{1}{2} + \big(\frac{3}{4}\big)^{10} < \frac{1}{2} + \frac{1}{16} = \frac{9}{16}$, $\big(\frac{3}{4}\big)^{10} < \frac{1}{16}$ being the probability of not obtaining an element of $AGL_n(\mathbb{F}_q)$ after ten trials. By repeating the protocol two times, it holds that no prover can fool the verifier into accepting $\underline{a_0} \not\equiv_{(AGL_n(\mathbb{F}_q), \phi_{\mathrm{IP1S}})} \underline{a_1}$ with a probability greater than $\big(\frac{9}{16}\big)^2 < \frac{1}{2}$. $\mathrm{PI}\big(AGL_n(\mathbb{F}_q), \phi_{\mathrm{IP1S}}\big)$ is then an interactive proof for the complementary language of $\mathrm{L}_{\mathrm{IP1S}}$ (i.e. $\{0,1\}^* \backslash \mathrm{L}_{\mathrm{IP1S}}$), where at most 4 messages are exchanged between the verifier and the prover. We don't detail the proof, but one can easily check that the same result holds for $\mathrm{PI}\big(AGL_n(\mathbb{F}_q) \times AGL_u(\mathbb{F}_q), \phi_{\mathrm{IP}}\big)$ and $\mathrm{PI}\big(GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q), \phi_{\mathrm{2PLE}}\big)$. (For IP, take $G = AGL_n(\mathbb{F}_q) \times AGL_u(\mathbb{F}_q), \mathcal{E} = \mathcal{M}_{n,n}(\mathbb{F}_q) \times \mathbb{F}_q^n \times \mathcal{M}_{u,u}(\mathbb{F}_q) \times \mathbb{F}_q^u$, and $C = 45$.)
The corollary then follows from a result of [7], stating that if the complementary of a language admits a constant round interactive protocol, then this language can not be NP-Complete, unless the polynomial hierarchy collapses. Hence, if the corresponding search problem is NP-Hard, then the polynomial hierarchy collapses, which is usually believed to be false. □

The new formalism introduced in this part allows then to upper bound the theoretical hardness of IP, 2PLE, and IP1S. More generally, it provides a new insight on the complexity of "IP-like" problems. The previous corollary can be indeed easily adapted to any instance of the Polynomial Equivalence problem. An "IP-like" problem is then intrinsically not NP-Hard. Furthermore, we believe that our formalism is of independent interest. It indeed procures a general framework for studying "IP-like" problems. This is however out of the scope of this paper. Let's now investigate another aspect of these problems.

4 An Algorithm for Solving 2PLE

We investigate here the practical hardness of a particular Polynomial Equivalence problem, namely 2PLE. Precisely, we present a new algorithm for solving this problem. We emphasize that – as explained in 1.1 – it is usually sufficient to consider this problem rather than its affine variant IP. Besides, any algorithm solving 2PLE can be transformed into an algorithm solving IP [11, 12]. Note that in the sequel, $(\underline{a}, \underline{b})$ will always refer to an element of $\mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$.

4.1 A First Attempt: Evaluation and Linearization

Instead of directly describing the details of our method, we present the different steps that yielded to this algorithm. Anyway, most of the intermediate results that we are going to present will be used in our final algorithm, but differently. Our earliest idea for solving 2PLE was based on the following remark. If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)U$, for $(S, U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$, then:

$$\underline{b}(\underline{p})U^{-1} = \underline{a}(\underline{p}S), \text{ for all } \underline{p} \in \mathbb{F}_q^n. \tag{1}$$

We hence obtain, for each $\underline{p} \in \mathbb{F}_q^n$, u non-linear equations in the $n^2 + u^2$ components of the matrices S and U^{-1}. We point out that the coefficients of U^{-1} only appear linearly in these equations. This is the advantage of considering the inverse of U rather than simply U in (1). The number of equations produced by (1) is significantly bigger that the number of unknowns. In this situation, one can simply use a linearization method[9] for solving the algebraic system. Unfortunately, our experiments rapidly revealed that the equations generated in this way are not all linearly independent. Besides, it also appeared that the number of unknowns is significantly bigger than the number of linearly independent equations. The use of a linearization method is then clearly no longer relevant. Let's explain this phenomenon.

Lemma 4.1. *Let $\underline{y} = (y_{1,1}, \ldots, y_{1,n}, \ldots, y_{n,1}, \ldots, y_{n,n})$, and $\underline{z} = (z_{1,1}, \ldots, z_{1,u}, \ldots, z_{u,1}, \ldots, z_{u,u})$. For each $i, 1 \leq i \leq u$, there exist a finite subset $S_i \subset \mathbb{F}_q^n$ and polynomials $p_\alpha^i \in \mathbb{F}_q[\underline{y}, \underline{z}]$, such that the following equality holds:*

$$\left(\underline{b}(\underline{x})U^{-1} - \underline{a}(\underline{x}S)\right)_i = \sum_{\alpha \in S_i} p_\alpha^i(S, U^{-1})\mathbf{x}^\alpha, \tag{2}$$

$p_\alpha^i(S, U^{-1})$ being the evaluation of p_α^i on $S = \{s_{i,j}\}_{1 \leq i,j \leq n}$ and $U^{-1} = \{u'_{i,j}\}_{1 \leq i,j \leq u}$, i.e.:

$$p_\alpha^i(S, U^{-1}) = p_\alpha^i(s_{1,1}, \ldots, s_{1,n}, \ldots, s_{n,1}, \ldots, s_{n,n}, u'_{1,1}, \ldots, u'_{1,u}, \ldots, u'_{u,u}, \ldots, u'_{u,u}).$$

Proof. Let's consider the components of S and U^{-1} as unknowns. The polynomials p_α^i are then obtained by symbolically developing $\left(\underline{b}(\underline{x})U^{-1} - \underline{a}(\underline{x}S)\right)_i$. Indeed, this last polynomial can be regarded as an element of

$$\mathbb{F}_q[s_{1,1}, \ldots, s_{1,n}, \ldots, s_{n,1}, \ldots, s_{n,n}, u'_{1,1}, \ldots, u'_{1,u}, \ldots, u'_{u,u}, \ldots, u'_{u,u}][x_1, \ldots, x_n], \tag{3}$$

i.e. a polynomial with unknowns $x_1, \ldots, x_n$ and whose coefficients are polynomials in the components of S and U^{-1}. In this setting, the polynomials p_α^i exactly correspond to the coefficients of the monomials (in $x_1, \ldots, x_n$) occurring in $\left(\underline{b}(\underline{x})U^{-1} - \underline{a}(\underline{x}S)\right)_i$. Lastly $S_i = \{\alpha \in \mathbb{F}_q^n : p_\alpha^i \neq 0\}$. $\square$

Remark 4.1. *– The cost of generating the polynomials p_α^i is proportional to the number of monomials occurring in $\left(\underline{b}(\underline{x})U^{-1} - \underline{a}(\underline{x}S)\right)_i$ viewed as a polynomial of (3), i.e. $O(n^{2D})$.*
– Each p_α^i is by construction the sum of a polynomial in $\underline{y}$, plus a linear polynomial in $\underline{z}$. Furthermore, the max. total degree reached by a monomial in the variables $\underline{y}$ is equal to D.

We observe that, for all $i, 1 \leq i \leq u$:

$$\left(\underline{b}(\underline{p})U^{-1} - \underline{a}(\underline{p}S)\right)_i = \sum_{\alpha \in S_i} p_\alpha^i(S, U^{-1}) p_1^{\alpha_1} \cdots p_n^{\alpha_n}, \text{for all } \underline{p} = (p_1, \ldots, p_n) \in \mathbb{F}_q^n,$$

It follows that, for all $\underline{p} \in \mathbb{F}_q^n$, the equations procured by (1) are linear combinations of the $p_\alpha^i(S, U^{-1})$. Besides, the number of polynomials p_α^i is limited by the number of monomials occuring in $\left(\underline{b}(\underline{p})U^{-1} - \underline{a}(\underline{p}S)\right)_i$. Thus, $u \cdot \mathrm{C}_{n+D}^D$ bounds from above the number of linearly independent equations provided by linearizing (1). On the other hand, the number of unknowns in the linearized system is equal to the number of monomials in the variables $\underline{y}$ of degree smaller than D, plus the u^2 variables[10] corresponding to $\underline{z}$. Using a rough bound, the linearization method yields to a linear system of at most $O(n^D)$ linearly independent equations with $O(n^{2D})$ unknowns.

[9] That is associating a new variable to each monomial of the system.

[10] By construction, the variables of $\underline{z}$ only appear linearly in the original system. We then no longer need to linearize the monomials in $\underline{z}$.

4.2 The 2PLE algorithm

The linearization can thus not be really employed for solving efficiently 2PLE. Still, Gröbner bases – which are an essential tool of commutative algebra – procure a method for solving algebraic systems. From a practical point of view, this approach seems quite promising. The system obtained by evaluating $\underline{b}(\underline{x})U^{-1} = \underline{a}(\underline{x}S)$ on several vectors is in effect overdetermined. Nevertheless, all the equations derived from $\underline{b}(\underline{p})U^{-1} = \underline{a}(\underline{p}S)$ are – according to (2) – linear combinations of the polynomials p^i_α previously introduced. It is hence sufficient (and more relevant) to only consider the system formed by the p^i_α. There is indeed no practical gain on considering systems composed of some linear combinations of p^i_α. Therefore:

Proposition 4.1. *Let $\mathcal{I} = \langle p^i_\alpha : \text{for all } i, 1 \leq i \leq u, \text{ and for all } \alpha \in S_i \rangle \subset \mathbb{F}_q[\underline{y}, \underline{z}]$ be the ideal generated by the polynomials p^i_α defined in lemma 4.1, and $V(\mathcal{I})$ be the following variety:*

$$V(\mathcal{I}) = \left\{ \underline{s} = (\underline{s_1}, \underline{s_2}) \in \mathbb{F}_q^{2n} : p^i_\alpha(\underline{s_1}, \underline{s_2}) = 0, \textit{for all } i, 1 \leq i \leq u, \textit{ and for all } \alpha \in S_i \right\}.$$

If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)U$, for some $(S, U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$, then:

$$\left(\phi_1(S), \phi_2(U^{-1})\right) \in V(\mathcal{I}),$$

with:

$$\begin{array}{l} \phi_1 : \mathcal{M}_{n,n}(\mathbb{F}_q) \rightarrow \mathbb{F}_q^{n^2}, S = \{s_{i,j}\}_{1 \leq i,j \leq n} \mapsto (s_{1,1}, \ldots, s_{1,n}, \ldots, s_{n,1}, \ldots, s_{n,n}), \textit{ and} \\ \phi_2 : \mathcal{M}_{u,u}(\mathbb{F}_q) \rightarrow \mathbb{F}_q^{u^2}, U' = \{u'_{i,j}\}_{1 \leq i,j \leq u} \mapsto (u'_{1,1}, \ldots, u'_{1,u}, \ldots, u'_{u,1}, \ldots, u'_{u,u}). \end{array}$$

Proof. The fact that $\underline{b} = \underline{a}(\underline{x}S)U$, implies that for all, $i, 1 \leq i \leq u$:

$$\left(\underline{b}(\underline{x})U^{-1} - \underline{a}(\underline{x}S)\right)_i = \sum_{\alpha \in S_i} p^i_\alpha(S, U^{-1})\mathbf{x}^\alpha = 0.$$

Thus, $p^i_\alpha(S, U^{-1}) = 0$, for all $i, 1 \leq i \leq u$, and for all $\alpha \in S_i$, i.e. $\left(\phi_1(S), \phi_2(U^{-1})\right) \in V(\mathcal{I})$. □

In other words, if $\underline{b} = \underline{a}(\underline{x}S)U$, for some $(S, U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$, then the variety $V(\mathcal{I})$ contains the components of the matrices S and U^{-1}. The system associated to $\mathcal{I}$ has $n^2 + u^2$ variables and its of degree D. Once again, we recall that the variables of $\underline{z}$ only appear linearly in this system. The number of equations of the system is equal to the number of monomials occurring in the polynomials of $\underline{a}$, i.e. $O\big(u \cdot \mathrm{C}^D_{n+D}\big)$. The system is then overdetermined.

Remark 4.2. *To guarantee that $V(\mathcal{I}) \subseteq \mathbb{F}_q^{2n}$, we must generally join the fields equations to the initial system. The fields considered in our case can be relatively large, leading then to a significant increase of the system's degree. This can artificially render impracticable the computation of a Gröbner basis. Fortunately, our systems are overdetermined and it is not necessary in practice to include the field equations. In our experiments the elements of $V(\mathcal{I})$ was indeed – without including these equations – all the times in $\mathbb{F}_q^{2n}$. It implies in particular that the hardness of 2PLE is not related to the size of the field. This is an important remark since the current security bound for 2PLE depends of this size.*

The next proposition is fundamental to understand the practical behaviour of our approach. This result permits furthermore to improve the efficiency of our method.

Proposition 4.2. *Let d be a positive integer, and $\mathcal{I}_d \subset \mathbb{F}_q[\underline{y}, \underline{z}]$ be the ideal generated by the polynomials p_α^i of maximal total degree smaller than d. Let also $V(\mathcal{I}_d)$ be the variety associated to $\mathcal{I}_d$. If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)U$, for some $(S,U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$, then:*

$$\big(\phi_1(S), \phi_2(U^{-1})\big) \in V(\mathcal{I}_d), \text{ for all } d, 0 \le d \le D,$$

ϕ_1 and ϕ_2 being defined as in proposition 4.1.

The proof is obviously deduced from the following result:

Property 4.1. *Let $(S,U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$, then the following equivalence holds:*

$$\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)U \Longleftrightarrow \underline{b}^{(d)}(\underline{x}) = \underline{a}^{(d)}(\underline{x}S)U, \text{ for all } d, 0 \le d \le D,$$

$\underline{b}^{(d)}$ $\big($resp. $\underline{a}^{(d)}\big)$ being the homogeneous components of degree d (i.e. the sum of the terms of total degree d) of the polynomials of $\underline{b}$ (resp. $\underline{a}$).

Proof. Let $\underline{b}'(\underline{x}) = \underline{b}(\underline{x})U^{-1}$. In [30], it has been proved that:

$$\underline{b}'(\underline{x}) = \underline{a}(\underline{x}S) \Longleftrightarrow \underline{b}'^{(d)}(\underline{x}) = \underline{a}^{(d)}(\underline{x}S)U, \text{ for all } d, 0 \le d \le D.$$

We conclude by remarking that $\underline{b}'^{(d)}(\underline{x}) = \underline{b}^{(d)}(\underline{x})U^{-1}$, for all $d, 0 \le d \le D$. □

Let's focus our attention on $\mathcal{I}_1$. The system associated to this ideal is only linear in the components of S and U^{-1}. Indeed, let $\underline{0_n}$ be the null vector of $\mathbb{F}_q^n$, and $A \in \mathcal{M}_{n,u}(\mathbb{F}_q)\big($resp. $B \in \mathcal{M}_{n,u}(\mathbb{F}_q)\big)$ be the matricial representation of $\underline{a}^{(1)}$ $\big($resp. $\underline{b}^{(1)}\big)$, i.e. $\underline{x}A = \underline{a}^{(1)}(\underline{x})$ $\big($resp. $\underline{x}B = \underline{b}^{(1)}(\underline{x})\big)$. According to property 4.1:

$$\underline{b} = \underline{a}(\underline{x}S)U, \text{ for some } (S,U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q) \Longrightarrow \begin{cases} \underline{b}^{(0)}(\underline{0_n})U^{-1} = \underline{a}^{(0)}(\underline{0_n}), \\ BU^{-1} = SA. \end{cases}$$

i.e. linear dependencies between the components S and U^{-1}. More precisely, we get $u(n+1)$ linear equations in the $n^2 + u^2$ components of the matrices solution. Anyway, we can not solve 2PLE just by using these equations. It is not necessary on the other hand to consider the system formed by all the p_α^i. According to proposition 4.2, we can actually restrict our attention to $\mathcal{I}_{d_0}$, with d_0 being the smaller integer rendering the system overdetermined. This d_0 can be defined in function of $\underline{a}$. Indeed, $d_0 = \min\{d > 1 : \underline{a}^{(d)} \neq \underline{0_u}\}$ (in practice, it is usually sufficient to take $d_0 = 2$). The hardness of an instance of 2PLE is then related to d_0 rather than to the maximal total degree D of this instance. It is also an important remark since the maximal degree of an instance is taken into account in the security estimate of 2PLE. Let's now describe our algorithm for solving this problem:

Input: $(\underline{a}, \underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$, `s.t.` $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)U$, `for some` $(S,U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$
`Let` $d_0 = \min\{d > 1 : \underline{a}^{(d)} \neq \underline{0_u}\}$
`1) Construct the` p_α^i `of max. total degree smaller than` d_0 `associated to` $(\underline{a}, \underline{b})$
`2) Compute` $V(\mathcal{I}_{d_0})$ `using the` F_5 `algorithm`
`3) Find an element of` $V(\mathcal{I}_{d_0})$ `corresponding to a solution of 2PLE`
Return `this solution`

The system associated to $\mathcal{I}_{d_0}$ is overdetermined by its very construction $\big(u^2 + n^2$ unknowns, and $O\big(u \cdot \mathrm{C}_{n+d_0}^{d_0}\big)$ equations$\big)$. The variety $V(\mathcal{I}_{d_0})$ is then very likely reduced to a solution of 2PLE (it has been indeed verified in our experiments). The complexity of this algorithm is then dominated by the Gröbner basis computation. It is really difficult to obtain a complexity bound really reflecting the practical behaviour of the F_5 algorithm. We therefore carry out now experimental results illustrating the practical efficiency of our approach.

4.3 Experimental Results

We present in this part experimental results obtained with our algorithm. Before that, let's provide the conditions our experiments.

4.4 Conditions of the Experiments

Generation of the instances
We have only considered instances $(\underline{a}, \underline{b})$ of 2PLE admitting a solution. We constructed the instances in the following way:
(1) Choose the polynomials of $\underline{a}$
(2) Randomly choose $(S, U) \in GL_n(\mathbb{F}_q) \times GL_u(\mathbb{F}_q)$
(3) Return the instance $\big(\underline{a}(\underline{x}), \underline{b}(\underline{x}) = \underline{a}(\underline{x}S)\big)$

Precisely, we constructed the polynomials of $\underline{a}$ in two different ways. The first one simply consists in randomly choosing – w.r.t. a given maximal total degree D – the polynomials[11] of $\underline{a}$. We shall call *random instance*, an instance of 2PLE generated in this manner. In the second method, $\underline{a}$ corresponds to the public key of a C* scheme [26]. An instance of 2PLE generated in this way will be named *C* instance*.

Programming language – Workstation
The experimental results have been obtained with an Opteron bi-processors 2.4 Ghz, with 8 Gb of Ram. The systems associated to an instance of 2PLE have been generated using the Magma software[12]. We used our proper implementation (in language C) of F_5 for computing the Gröbner bases. For the sake of comparison, we used however sometimes the last version of Magma (i.e. 2.12) for obtaining these bases. This version indeed includes an implementation of the F_4 algorithm.

Tables Notations
The following notations are used in the tables below:
– n, the number of variables,
– q, the size of the field,
– deg, the maximal total degree of the instance considered,
– T_{Gen}, the time needed to construct the system,
– T_{F_5}, the time of our algorithm for finding a solution of 2PLE (using the F_5 algorithm for computing the Gröbner bases),
– T, the total time of our algorithm, i.e. $T = T_{F_5} + T_{Gen}$,
– T_{Mag}, the time of our algorithm for recovering a solution of 2PLE (using Magma v. 2.12 for computing the Gröbner bases),
– $q^{n/2}$ (resp. q^n), the security bound [11, 12] for instances of $deg = 2$ (resp. $deg > 2$).

4.5 Practical Results – Random Instances

Let's now present the results obtained on random instances of 2PLE. We emphasize that this family of instances is employed in the authentication and signature schemes based on 2PLE proposed by J. Patarin at Eurocrypt'96 [27, 28]. Therefore – since our main motivation is to

[11] Precisely, each polynomial is a random linear combination of all the monomials of total degree smaller (or equal) to D. Note that we obtain in this way dense polynomials.
[12] `http://magma.maths.usyd.edu.au/magma/`

study the security of these schemes – we can w.l.o.g. restrict our attention to the $u = n$ case. It is indeed only these instances that J. Patarin (see [27, 28]) suggested to use in practice.

n	q	deg	T_{Gen}	T_{F_5}	T_{Mag}/T_{F_5}	T	$q^{n/2}$
8	2^{16}	2	0.35 s.	0.14 s.	6	0.49 s.	2^{64}
10	2^{16}	2	1.66 s.	0.63 s.	10	2.29 s.	2^{80}
12	2^{16}	2	7.33 s.	2.16 s.	16	9.49 s.	2^{96}
15	2^{16}	2	48.01 s.	10.9 s.	23	58.91 s.	2^{120}
17	2^{16}	2	137.21 s.	27.95 s.	31	195.16 s.	2^{136}
20	2^{16}	2	569.14 s.	91.54 s.	41	660.68 s.	2^{160}
10	65521	2	1.21 s.	0.44 s.	10	1.65 s.	$\approx 2^{80}$
15	65521	2	35.58 s.	8.08 s.	23	43.66 s.	$\approx 2^{120}$
20	65521	2	434.96 s.	69.96 s.	41	504.92 s.	$\approx 2^{160}$
23	65521	2	1578.6 s.	235.92 s.		1814 s.	$\approx 2^{184}$

Remark 4.3. *– Our implementation of F_5 is faster than the Gröbner basis algorithm available in Magma 2.12. For $n = 20$, F_5 is for instance 41 times faster than Magma.*
– To fix the ideas, $u = n = 8$, and $u = n = 16$ were two challenges proposed at Eurocrypt[28].
– We obtained exactly the same results for random instances of $deg > 2$. On the other hand, the security estimate for these instances is at least equal to $2^{128}(n = 8)$. The maximal total degree of the systems is indeed the same than for instances of $deg = 2$, i.e. d_0 is equal to 2 independently of D. In other words, increasing the maximal total degree of a random instance will not change its practical hardness. We observe the same behaviour for the size of the field, that is increasing q not really change the hardness of a random instance. This will indeed modify only the cost of the arithmetic operations in the different steps our algorithm.

Interpretation of the results
In all these experiments, the varieties computed was reduced to one element, i.e. the components of the matrices solution of 2PLE. Furthermore, we observe in practice that the complexity of our algorithm is dominated by the time required to construct the system, and not by the Gröbner basis computation. This is surprising, but it clearly highlights that the systems considered here can be easily solved in practice. The generation of the systems being polynomial, we then conclude experimentally that our algorithm solves random instances of 2PLE in polynomial-time. This conclusion is supported by the fact that – in all these experiments – F_5 generated matrices of size at most equal to n^3. We hence obtain experimentally a complexity of n^9 for computing the Gröbner bases associated to random instances of 2PLE. Let's look how our algorithm behaves on C* instances.

4.6 Practical Results – C* Instances

We now present the results obtained on C* instances $(\underline{a}, \underline{b})$ of $deg = D$. We highlight that these instances are used in the traitor tracing scheme described in [5]. In this context, we also have $u = n$. The polynomials of $\underline{a}$ indeed correspond to the public-key of a C* scheme [26]. Precisely, these polynomials are the "multivariate representation" of a univariate monomial (see [5] for details concerning the generation of this multivariate representation). The univariate polynomial has the following shape: $m^{1+q^{\theta_1}+q^{\theta_2}+\cdots+q^{\theta_{D-1}}}$, with $\theta_1, \theta_2, \cdots, \theta_{D-1} \in \mathbb{N}^*$.

n	q	deg	T_{Gen}	T_{F_5}	T_{Mag}/T_{F_5}	T	q^n
5	2^{16}	4	0.2 s.	0.13 s.	45	0.33 s.	2^{80}
6	2^{16}	4	0.7 s.	1.03 s.	64	1.73 s.	2^{96}
7	2^{16}	4	1.5 s.	6.15 s.	90	7.65 s.	2^{112}
8	2^{16}	4	3.88 s.	54.34 s.	112	58.22 s.	2^{128}
9	2^{16}	4	5.43 s.	79.85 s.	145	85.28 s.	2^{144}
10	2^{16}	4	12.9 s.	532.33 s.	170	545.23 s.	2^{160}

Remark 4.4. *– $n = 5$, and $deg = 4$ is the first challenge proposed at Asiacrypt'03 [5].*
– Similarly to random instances, we observed that the size of the field and the degree D do not really change the practical hardness of the C instances. (For instance, we only solved here systems of degree 2, i.e. $d_0 = 2$.) We conclude that it is a general behaviour of 2PLE instances.*

Interpretation of the results
Our algorithm is no longer polynomial for C* instances. The systems obtained for these instances are indeed harder to solve than the random ones. We believe that it is due to the fact that the systems are here sparser. The equality $\underline{b}(\underline{0_n}) = \underline{a}(\underline{0_n})U$ does not provide any information. Indeed, $\underline{b}(\underline{0_n}) = \underline{a}(\underline{0_n}) = \underline{0_n}$ in the C* case. It is not clear yet but it seems that C* instances with $n = 19$ (the second challenge proposed in [5]), can not be solved with our approach. More generally, we think that $d_{min} = \min\{d \leq 0 : \underline{a}^{(d)} \neq \underline{0_u}\}$ provides a relevant measure of the practical hardness of 2PLE instances. It seems actually that this practical difficulty increases in function of d_{min}. Indeed, for random instances of 2PLE, $d_{min} = 0$ and our algorithm is polynomial. For C* instances, $d_{min} = 1$ and we change of complexity class. We also checked that the practical complexity increases for homogeneous instances of degree 2, i.e. $d_{min} = 2$.
To summarize, for $d_{min} = 0$ it is relatively clear that our algorithm is polynomial. For $d_{min} \geq 1$, we guess that our algorithm is subexponential in n, and will depend of d_{min}. This anyway needs further investigations. It is indeed an open problem to determine – in function of d_{min} – the asymptotical complexity of our algorithm. It could be possible that techniques presented in [3, 4] provides an answer.

5 Conclusion

The contribution of this paper is twofold. We introduced a new formalism providing a unified point of view of Polynomial Equivalence problems. We believe that this formalism is of independent interest. Anyway, it permitted to upper bound the theoretical hardness of IP, 2PLE, IP1S, and more generally on "IP-like" problems. We also presented a simple and efficient (i.e. in the sense that we break several challenges proposed in the literature) algorithm for solving 2PLE, which is one of the main problems in multivariate cryptography. To conclude, we mention that it is very likely that our algorithm could be adapted for attacking schemes such as Sflash [13] and 2R [24]. The security of these systems are indeed closely related to the problems addressed in this paper.

References

[1] W.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] 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.

[4] 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.

[5] O. Billet, and H. Gilbert. *A Traceable Block Cipher.* Advances in Cryptology – ASIACRYPT 2003, Lecture Notes in Computer Science, vol. 2894, Springer–Verlag, pp. 331-346, 2003.

[6] A. Biryukov, C. De Cannière, A. Braeken, and B. Preneel. *A Toolbox for Cryptanalysis: Linear and Affine Equivalence Algorithms.* Advances in Cryptology – EUROCRYPT 2003, Lecture Notes in Computer Science, vol. 2656, Springer–Verlag, pp. 33-50, 2003.

[7] R. B. Boppana, J. Hastad, and S. Zachos. *Does* co–NP *Have Short Interactive Proofs?* Information Processing Letters, 25(2), pp. 127–132, 1987.

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

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

[10] N. Courtois. *La sécurité des primitives cryptographiques basées sur des problèmes algébriques multivariables: MQ, IP, MinRank, HFE.* Ph.D. Thesis, Paris, 2001.

[11] N. Courtois, L. Goubin, and J. Patarin. *Improved Algorithms for Isomorphism of Polynomials.* Advances in Cryptology - EUROCRYPT 1998, Lecture Notes in Computer Science, vol. 1403, Springer-Verlag, pp. 84–200, 1998.

[12] N. Courtois, L. Goubin, and J. Patarin. *Improved Algorithms for Isomorphism of Polynomials - Extended Version.* Available from `http://www.minrank.org`.

[13] 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`.

[14] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations.* Advances in Cryptology – EUROCRYPT 2000, Lecture Notes in Computer Science, vol. 1807, Springer–Verlag, pp. 392-407, 2000.

[15] 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.

[16] J.-C. Faugère. *A New Efficient Algorithm for Computing Gröbner Bases (F_4).* Journal of Pure and Applied Algebra, 139(1-3), pp. 61–88, June 1999.

[17] 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.

[18] J.-C. Faugère, and A. Joux. *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems using Gröbner bases.* Advances in Cryptology - CRYPTO 2003, Lecture Notes in Computer Science, vol. 2729, Springer-Verlag, pp. 44–60, 2003.

[19] 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.

[20] P. Felke *On certain Families of HFE-type Cryptosystems.* Proceedings of WCC05, International Workshop on Coding and Cryptography, March 2005.

[21] S. Fortin. *The Graph Isomorphism problem.* Technical Report 96-20, University of Alberta, 1996.

[22] M. R. Garey, and D. B. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[23] W. Geiselmann, R. Steinwandt, and T. Beth. *Attacking the Affine Parts of SFLASH.* Cryptography and Coding, 8th IMA International Conference, vol. 2260, Springer–Verlag, pp. 355-359, 2001.

[24] 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,

[25] M. Hoffman. *Group-theoretic algorithms and Graph Isomorphism.* Lecture Notes in Computer Science, vol. 136, Springer–Verlag, 1982.

[26] 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.

[27] J. Patarin. *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms.* Advances in Cryptology – EUROCRYPT 1996, Lecture Notes in Computer Science, vol. 1070, Springer-Verlag, pp. 33–48,1996.

[28] J. Patarin. *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms – Extended Version.* Available from `http://www.minrank.org/hfe/`.

[29] 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.

[30] L. Perret. *A Fast Cryptanalysis of the Isomorphism of Polynomials with One Secret Problem.* Advances in Cryptology - EUROCRYPT 2005, Lecture Notes in Computer Science, vol. 3494, pp. 354–370, 2005.