A Fast Cryptanalysis of the Isomorphism of Polynomials with One Secret Problem

Ludovic Perret

ENSTA, UMA
32 Boulevard Victor, 75739 Paris Cedex 15, France
lperret@ensta.fr

Abstract. At Eurocrypt'96, Patarin proposed [9] new cryptographic schemes based on the *Isomorphism of Polynomials with one Secret* problem (IP1S) [9]. We study in this paper a restriction of IP1S called *Polynomial Linear Equivalence* problem (PLE) [7]. We show that PLE is in fact not a restriction of IP1S, in the sense that any algorithm solving PLE can be efficiently transformed into an algorithm for solving IP1S. Motivated by the cryptanalysis of schemes based on IP1S, we present a new efficient algorithm for solving PLE. This algorithm is mainly based on a differential property of PLE. The main advantage of this approach is to translate PLE into a simple linear algebra problem. The performances of our algorithm evidence that, with the parameters proposed in [9], schemes based on IP1S are far from achieving the security level required for cryptographic applications.

Keywords: Cryptanalysis, Isomorphism of Polynomials with One Secret (IP1S), Polynomial Linear Equivalence (PLE), Jacobian Matrix.

1 Introduction

IP1S has been originally introduced by Patarin [9] to circumvent the problem of practicality encountered when using the *Graph Isomorphism* problem as an underlying problem for zero-knowledge authentication protocols [4].
IP1S can be outlined as follows: given multivariate polynomials $\big(a_1(x_1\ldots,x_n),\ldots,a_u(x_1\ldots,x_n)\big)$ and $\big(b_1(x_1\ldots,x_n),\ldots,b_u(x_1\ldots,x_n)\big)$ over $\mathbb{F}_q[x_1,\ldots,x_n]$, find - if any - an invertible matrix $S\in GL_n(\mathbb{F}_q)$ and a vector $\underline{T}\in\mathbb{F}_q^n$, such that:

$$b_i(x_1\ldots,x_n)=a_i\big((x_1\ldots,x_n)S+\underline{T}\big),\text{for all } i, 1\leq i\leq u.$$

In other words, Graphs have been replaced by multivariate polynomials and permutations by bijective affine mappings. A new authentication protocol, based on IP1S, as well as a public key signature scheme were then designed in [9]. The main motivation of this paper is to study, from both a theoretical and practical point of view, the security of these schemes. To do so, we address here a relevant variant of it. The problem we call *Polynomial Linear Equivalence* problem (PLE) [7], which is the restriction of IP1S to bijective linear mappings. We stress that this is in fact not a restriction since we prove in this paper that IP1S and PLE are equivalent, in the sense that any algorithm solving PLE can be efficiently transformed into an algorithm for solving IP1S.

1.1 Previous Work

To the best of our knowledge, the first algorithm presented for IP1S is due to Geiselmann, Meier and Steinwandt [3]. We here briefly recall its principle and refer the reader to the original paper for a detailed description.
Let $\big((a_1,\ldots,a_u),(b_1,\ldots,b_u)\big) \in \mathbb{F}_q[x_1\ldots,x_n]^u \times \mathbb{F}_q[x_1\ldots,x_n]^u$, and $(S,\underline{T}) \in GL_n(\mathbb{F}_q)\times\mathbb{F}_q^n$ such that:

$$b_i(x_1\ldots,x_n) = a_i\big((x_1\ldots,x_n)S+\underline{T}\big),\ \text{for all } i, 1\leq i\leq u.$$

Moreover, let $\underline{e_j}\in\mathbb{F}_q^n$ be the vector with its jth component equal to one and zero otherwise. The main idea is to remark that if $\underline{\ell_j}\in\mathbb{F}_q^n$ is the jth row of the matrix S, then:

$$b_i(\underline{e_j}) = a_i\big(\underline{\ell_j}+\underline{T}\big),\ \text{for all } i, 1\leq i\leq u.$$

When $\underline{T}\in\mathbb{F}_q^n$ is given, an exhaustive search among $\mathbb{F}_q^n$ is then performed to recover:

$$L_j = \{\underline{\ell}\in\mathbb{F}_q^n : b_i(\underline{e_j}) = a_i(\underline{\ell}+\underline{T}),\ \text{for all } i, 1\leq i\leq u\},$$

which is a set of candidate vectors for the jth row of S.
Soon after, Levy-dit-Vehel and Perret in [7] have remarked that the jth row of S is a zero of the following system of non-linear equations:

$$\big\{a_1(\underline{x}+\underline{T})-b_1(\underline{e_j})=0,\cdots,a_u(\underline{x}+\underline{T})-b_u(\underline{e_j})=0\big\}. \qquad (1)$$

Therefore, the set L_j of candidates for the jth row of S is equal to the set of zeroes of (1). Hence, they have substituted the exhaustive search of the elements of L_j by the computation of a Gröbner basis [7]. In this work, we use very basic tools of linear algebra for solving IP1S.

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 more formally the PLE and IP1S problems, which are the main concern of this paper.
In Section 3, we prove that PLE is equivalent to IP1S, i.e. any algorithm solving PLE can be efficiently transformed into an algorithm for solving IP1S.
In Section 4, differential properties of PLE are presented. These properties give a strong relation between the Jacobian matrices of an instance of PLE and solutions of this problem. We also show that structural properties of PLE can be used to obtain linear equations in the components of a solution of PLE.
A new algorithm for solving PLE is described in Section 5. Using properties of section 4, we show that a partial knowledge of a solution allows us to recover

it entirely by solving a suitable linear system of equations. It appears that the algorithm presented in this section is much more efficient than algorithms previously proposed [3, 7]. This is illustrated in the last part of this paper by giving experimental results obtained with our algorithm.

2 Preliminaries

2.1 Notations

We introduce in this part the notations used throughout this paper. We denote by $\mathbb{F}_q$, the finite field with $q = p^r$ elements (p a prime, and $r \geq 1$), by $\underline{x}$ the vector $(x_1, \ldots, x_n)$, by $\mathbb{F}_q[\underline{x}] = \mathbb{F}_q[x_1, \ldots, x_n]$, the polynomial ring in the n indeterminate $x_1, \ldots, x_n$ over $\mathbb{F}_q$, and $f(\underline{x})$ stands for $f(x_1, \ldots, x_n)$. Moreover, let g and $h_1, \ldots, h_n$ be polynomials of $\mathbb{F}_q[\underline{x}]$; by $g \circ \underline{h}$ we shall mean the functional composition $g\big(h_1, \cdots, h_n\big)$ of g and the h_i's.
A *monomial* is a power product of the variables $x_1, \ldots, x_n$, and a *term* is a coefficient multiplied by a monomial. We shall define the *total degree* of a monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}, (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, by the sum $\sum_{i=1}^n \alpha_i$. Obviously, the *total degree* of a term $cx_1^{\alpha_1} \cdots x_n^{\alpha_n}, c \in \mathbb{F}_q^*$, is the total degree of $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. The *leading term* of f is the largest term among the terms of f w.r.t. some admissible ordering on the monomials. For example, the lexicographical order $\prec_{LEX}$, defined by:

$$x_1^{\alpha_1} \cdots x_n^{\alpha_n} \prec_{LEX} x_1^{\beta_1} \cdots x_n^{\beta_n} \Longleftrightarrow \begin{cases} \text{the first coordinates } \alpha_i \text{ and } \beta_i \text{ from the left} \\ \text{which are different satisfy } \alpha_i < \beta_i, \end{cases}$$

is an admissible order.
Let $f \in \mathbb{F}_q[\underline{x}]$, the *degree* of f is the total degree of its leading term. We shall say that f is *homogeneous* of degree d if every term appearing in f has total degree d. An important fact is that every polynomial can be written uniquely as a sum of homogeneous polynomials. Namely $f = \sum_d f^{(d)}$, with $f^{(d)}$ being the sum of all terms of f of total degree d. Notice that each $f^{(d)}$ is homogeneous, and we call $f^{(d)}$ the dth *homogeneous component of* f. If f is of maximal total degree d, we shall call *homogenization* of f, denoted by F, the polynomial:

$$F(x_1, \ldots, x_n, z) = \sum_{i=0}^{d} f^{(i)}(x_1, \ldots, x_n) z^{d-i}. \tag{2}$$

The polynomials f and F are related in the following way:

$$F(\underline{x}, z) = z^d f\big(\frac{x_1}{z}, \ldots, \frac{x_n}{z}\big) = z^d f\big(\frac{\underline{x}}{z}\big). \tag{3}$$

Evaluating F in $(\underline{x}, 1)$ yields f, i.e. $F(\underline{x}, 1) = f(\underline{x})$. This process is called *dehomogenization.*
We extend now some of the notations previously given to vectors of polynomials. Precisely, for $\underline{a} = \big(a_1, \cdots, a_u\big) \in \mathbb{F}_q[\underline{x}]^u$, we shall denote by $\underline{a}^{(d)} = (a_1^{(d)}, \ldots, a_u^{(d)})$ the dth homogeneous components of the polynomials of $\underline{a}$.

We shall denote by $\mathcal{M}_{n,u}(\mathbb{F}_q)$ the set of $n \times u$ matrices whose components are in $\mathbb{F}_q$. For $M \in \mathcal{M}_{n,u}(\mathbb{F}_q)$, we set $Ker(M) = \{\underline{x} \in \mathbb{F}_q^n : \underline{x}M = \underline{0_u}\}$, $\underline{0_u}$ being the null vector of $\mathbb{F}_q^u$. As usual, $GL_n(\mathbb{F}_q)$ denotes the set of invertible matrices of $\mathcal{M}_{n,n}(\mathbb{F}_q)$, and we denote by $AGL_n(\mathbb{F}_q)$ the cartesian product $GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$.

2.2 Jacobian matrix

Let $f = \sum_i a_i x^i \in \mathbb{F}_q[x]$, the *formal derivative* of f is the polynomial $\frac{df}{dx} = \sum_i i a_i x^{i-1} \in \mathbb{F}_q[x]$. More generally, when $f \in \mathbb{F}_q[x_1, \ldots, x_n]$, the *partial derivatives* of f, denoted by $\frac{\partial f}{\partial x_i}$, $1 \le i \le n$, are defined by considering f as a polynomial in x_i with coefficients in $\mathbb{F}_q[x_1 \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$. It is not hard to check that the $\partial/\partial x_i$'s commute with one another.

Definition 1. *The* Jacobian matrix *of* $\underline{f} = (f_1, \ldots, f_u) \in \mathbb{F}_q[\underline{x}]^u$, *denoted by* $J_{\underline{f}}(\underline{x})$, *is the* $u \times n$ *matrix whose components are the partial derivatives of the polynomials of* $\underline{f}$, *i.e.:*

$$J_{\underline{f}}(\underline{x}) = \left\{ \frac{\partial f_i}{\partial x_j}(\underline{x}) \right\}_{1 \le j \le n}^{1 \le i \le u}$$

The property of partial derivatives that we use in this paper is the chain rule condition:

$$\frac{\partial (g \circ \underline{h})}{\partial x_i}(\underline{x}) = \sum_{j=1}^{n} \frac{\partial g}{\partial x_j}(\underline{h}(\underline{x})) \frac{\partial h_j}{\partial x_i}(\underline{x}), \text{for all } i, 1 \le i \le n.$$

2.3 The IP1S and PLE problems

Let $\big(\underline{a} = (a_1, \ldots, a_u), \underline{b} = (b_1, \ldots, b_u)\big) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$. We shall say that $(\underline{a}, \underline{b})$ are *affine-equivalent*, denoted by $\underline{a} \equiv_A \underline{b}$, if there exists $(S, \underline{T}) \in AGL_n(\mathbb{F}_q)$, s.t.:

$$b_i(x_1 \ldots, x_n) = a_i\big((x_1 \ldots, x_n)S + \underline{T}\big), \text{for all } i, 1 \le i \le u.$$

We call such a pair an *affine equivalence pair*. The *Isomorphism of Polynomials with one Secret* problem (IP1S) is then the one of finding - if any - an affine equivalence pair between the polynomials of $\underline{a}$ and $\underline{b}$. We mention that this problem is also called *Polynomial Affine Equivalence* problem (PAE) in [7].
A natural variant of this problem is to consider linear bijective mappings.
We shall say that $(\underline{a}, \underline{b})$ are *linear-equivalent*, denoted by $\underline{a} \equiv_L \underline{b}$, if there exists $S \in GL_n(\mathbb{F}_q)$, such that:

$$b_i(\underline{x}) = a_i(\underline{x}S), \text{for all } i, 1 \le i \le u. \tag{4}$$

In the sequel we shall denote, for convenience, equations (4) by $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$.
We call the matrix S a *linear equivalence matrix*. The *Polynomial Linear Equivalence* problem (PLE) is then the one of finding - if any - a linear equivalence matrix between $\underline{a}$ and $\underline{b}$.

3 IP1S and PLE are equivalent

Before giving our complexity results, we need to present structural properties of PLE and IP1S.

Property 1 *If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S + \underline{T})$, for some $(S, \underline{T}) \in AGL_n(\mathbb{F}_q)$, then:*

$$b_i^{(D_i)}(\underline{x}) = a_i^{(D_i)}(\underline{x}S), \textit{for all } i,\ 1 \leq i \leq u,$$

D_i being, for all i, $1 \leq i \leq u$, the degree of the homogeneous component of highest degree of b_i.

Proof. For all i, $1 \leq i \leq u$, $b_i(\underline{x}) = a_i(\underline{x}S + \underline{T})$, for some $(S, \underline{T}) \in AGL_n(\mathbb{F}_q)$ implies that $b_i(\underline{x} - \underline{T}S^{-1}) = a_i(\underline{x}S)$. We stress that $b_i^{(D_i)}(\underline{x} - \underline{T}S^{-1})$, which is the homogeneous component $b_i^{(D_i)}$ of b_i evaluated in $\underline{x} - \underline{T}S^{-1}$, contains the terms of total degree D_i of $b_i(\underline{x} - \underline{T}S^{-1})$.
Indeed, let $b_i^{(D_i)}(\underline{x}) = \sum_{1 \leq j_1, \ldots, j_{D_i} \leq n} b_{i,j_1,\ldots,j_{D_i}}^{(D_i)} x_{j_1} \cdots x_{j_{D_i}}$, be the homogeneous component of degree D_i of b_i. Since:

$$\prod_{k=1}^{D_i} \left(x_{j_k} - (\underline{T}S^{-1})_{j_k}\right) = \underbrace{x_{j_1} \cdots x_{j_{D_i}}}_{\text{total degree } D_i} + \text{terms of total degree} < D_i.$$

We have:

$$\begin{aligned} b_i^{(D_i)}(\underline{x} - \underline{T}S^{-1}) &= \sum_{1 \leq j_1, \ldots, j_{D_i} \leq n} b_{i,j_1,\ldots,j_{D_i}}^{(D_i)} \prod_{k=1}^{D_i} \left(x_{j_k} - (\underline{T}S^{-1})_{j_k}\right) \\ &= \underbrace{b_i^{(D_i)}(\underline{x})}_{\text{total degree } D_i} + \text{terms of total degree} < D_i. \end{aligned}$$

Finally, by equating the terms of total degree D_i of $b_i(\underline{x} - \underline{T}S^{-1})$ with those of $a_i(\underline{x}S)$, we get that $b_i^{(D_i)}(\underline{x}) = a_i^{(D_i)}(\underline{x}S)$, for all $i, 1 \leq i \leq u$. □

Remark 1 *Let $\left(\underline{a} = (a_1, \ldots, a_u), \underline{b} = (b_1, \ldots, b_u)\right) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$. In the rest of the paper, D_i will always denote the degree of the homogeneous component of highest degree of b_i. Moreover, we set $D = max_{1 \leq i \leq u}(D_i)$.*

We now give the linear counterpart of property 1. Remark that the next property already appeared in [7], but is quoted here for the sake of completeness.

Property 2 *Let $S \in GL_n(\mathbb{F}_q)$, we have:*

$$\underline{b}(\underline{x}) = \underline{a}(\underline{x}S) \Longleftrightarrow \underline{b}^{(d)}(\underline{x}) = \underline{a}^{(d)}(\underline{x}S), \textit{for all } d, 0 \leq d \leq D.$$

Proof. Let $S \in GL_n(\mathbb{F}_q)$, such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$. For each $i, 1 \leq i \leq u$, and for all d, $0 \leq d \leq D$, the terms of total degree d of $a_i(\underline{x}S)$ are equal to those of the homogeneous polynomial $a_i^{(d)}$ evaluated in $\underline{x}S$, i.e. the terms of $a_i^{(d)}(\underline{x}S)$. Thus, by equating the terms of total degree d of $b_i(\underline{x})$ with those of $a_i(\underline{x}S)$, we get that for all $i, 1 \leq i \leq u$:

$$b_i^{(d)}(\underline{x}) = a_i^{(d)}(\underline{x}S), \text{for all } d, 0 \leq d \leq D.$$

Let $S \in GL_n(\mathbb{F}_q)$ and suppose now that for all $i, 1 \leq i \leq u, b_i^{(d)}(\underline{x}) = a_i^{(d)}(\underline{x}S)$, for all $d, 0 \leq d \leq D$. Consequently, we get that $\sum_{d=0}^{D} b_i^{(d)}(\underline{x}) = \sum_{d=0}^{D} a_i^{(d)}(\underline{x}S)$, i.e. $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$. □

We now introduce some additional notations. We shall call dPLE (resp. dIP1S) the decisional version of PLE (resp. IP1S); that is, the problem of deciding whether $(\underline{a}, \underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$ are linear-equivalent (resp. affine-equivalent). Finally, we would like to recall that a polynomial-time many-one reduction (also known as Karp reduction) is defined as follows:

Definition 2. *[5] Let A and B be two decisional problems. A is* polynomial-time many-one reducible *to B, denoted by $A \leq_p^m B$, iff there exists a polynomial-time computable function f, such that for any instance x of A, we have:*

$$x \in L_A \Longleftrightarrow f(x) \in L_B,$$

L_A and L_B being the set of YES instances of A and B.
Moreover, A and B are polynomial-time many-one equivalent*, denoted by $A \equiv_p^m B$, iff $A \leq_p^m B$ and $B \leq_p^m A$.*

For dIP1S and dPLE, we have the following (surprising) result:

Proposition 1 *dIP1S is polynomial-time many-one reducible to dPLE.*

Proof. In order to prove that dIP1S$\leq_p^m$dPLE, we define a function $f : \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u \to \mathbb{F}_q[\underline{x}, z]^{u+1} \times \mathbb{F}_q[\underline{x}, z]^{u+1}$ as follows. For all $(\underline{a}, \underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$:

$$f\big(\underline{a}(\underline{x}), \underline{b}(\underline{x})\big) = \big(\underline{A}(\underline{x}, z), \underline{B}(\underline{x}, z)\big),$$

with $\underline{A}(\underline{x}, z) = \big(A_1(\underline{x}, z), \ldots, A_u(\underline{x}, z), z\big)$ and $\underline{B}(\underline{x}, z) = \big(B_1(\underline{x}, z), \ldots, B_u(\underline{x}, z), z\big)$. The A_i's (resp. B_i's) being the homogenizations of the a_i's (resp. b_i's). One can see at once that, according to (2), f can be computed in polynomial-time.
Now, let $(\underline{a}, \underline{b}) \in L_{dIP1S}$, i.e. $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S + \underline{T})$, for some $\big(S = \{s_{i,j}\}_{1 \leq i,j \leq n}, \underline{T} = (t_1, \ldots, t_n)\big) \in AGL_n(\mathbb{F}_q)$. From this affine equivalence pair, we define the following matrix:

$$S' = \begin{pmatrix} s_{1,1} & s_{1,2} & \ldots & s_{1,n} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{n,1} & s_{n,2} & \ldots & s_{n,n} & 0 \\ t_1 & t_2 & \ldots & t_n & 1 \end{pmatrix}.$$

We mention that since $S \in GL_n(\mathbb{F}_q)$, then $S' \in GL_{n+1}(\mathbb{F}_q)$. Indeed, it's inverse is $\begin{pmatrix} S^{-1} & {0_n}^t \\ -TS^{-1} & 1 \end{pmatrix}$. Moreover, we have:

$$\begin{aligned} (\underline{x}, z)S' &= \left(\sum_{j=1}^n x_j s_{j,1} + t_1 z, \ldots, \sum_{j=1}^n x_j s_{j,n} + t_n z, z \right) \\ &= \left(\underline{x}S + \underline{T}z, z\right) \end{aligned} \tag{5}$$

Recall that for all $i, 1 \le i \le u$, D_i denotes the degree of the homogeneous component of highest degree of b_i. Note that for all $z \neq 0$, $z^{D_i} b_i(\frac{\underline{x}}{z}) = z^{D_i} a_i(\frac{\underline{x}}{z}S + \underline{T})$. Thus, using (3) and (5), we get that for all $i, 1 \le i \le u$:

$$B_i\left(\underline{x}, z\right) = z^{D_i} a_i\left(\frac{\underline{x}}{z}S + \underline{T}\right) = z^{D_i} a_i\left(\frac{\underline{x}S + \underline{T}z}{z}\right) = A_i\left(\underline{x}S + \underline{T}z, z\right) = A_i\left((\underline{x}, z)S'\right).$$

To handle the case $z = 0$, we use property 1. According to it, we know that if $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S + \underline{T})$, for some $(S, \underline{T}) \in AGL_n(\mathbb{F}_q)$, then $b_i^{(D_i)}(\underline{x}) = a_i^{(D_i)}(\underline{x}S)$, for all i, $1 \le i \le u$. Therefore, for $z = 0$, and for all $i, 1 \le i \le u$:

$$A_i\left((\underline{x}, 0)S'\right) = A_i\left(\underline{x}S, 0\right) = a_i^{(D_i)}\left(\underline{x}S\right) = b_i^{(D_i)}\left(\underline{x}\right) = B_i\left(\underline{x}, 0\right).$$

Finally, we remark that $A_{u+1}\left((\underline{x}, z)S'\right) = A_{u+1}\left(\underline{x}S + \underline{T}z, z\right) = z = B_{u+1}\left(\underline{x}, z\right)$. Thus, we get that $f\left(\underline{a}, \underline{b}\right) = \left(\underline{A}, \underline{B}\right) \in L_{dPLE}$.

Now, let $f\left(\underline{a}, \underline{b}\right) = \left(\underline{A}, \underline{B}\right) \in L_{dPLE}$, i.e. $\underline{B}\left(\underline{x}, z\right) = \underline{A}\left((\underline{x}, z)S''\right)$, for some $S'' = \{s''_{i,j}\}_{1 \le i,j \le n+1} \in GL_{n+1}(\mathbb{F}_q)$. Due to the particular shape of the polynomials of $\underline{A}$ and $\underline{B}$, we must have $z = \sum_{j=1}^n x_j s''_{j,n+1} + z s''_{n+1,n+1}$, i.e. $s''_{j,n+1} = 0$, for all $j, 1 \le j \le n$ and $s''_{n+1,n+1} = 1$. Thus, the linear equivalence matrix S'' must leave z unchanged. Therefore, if we set $h_1(S'') = \{s''_{i,j}\}_{1 \le i,j \le n}$ and $h_2(S'') = (s''_{n+1,1}, \ldots, s''_{n+1,n})$ then for all $i, 1 \le i \le u$, we have:

$$B_i\left(\underline{x}, z\right) = A_i\left((\underline{x}, z)S''\right) = A_i\left(\underline{x}h_1(S'') + zh_2(S''), z\right) = z^{D_i} a_i\left(\frac{\underline{x}}{z}h_1(S'') + h_2(S'')\right).$$

For $z = 1$, we get in particular that:

$$\underline{B}(\underline{x}, 1) = \left(\underline{b}(\underline{x}), 1\right) = \underline{A}((\underline{x}, 1)S'') = \left(\underline{a}(\underline{x}h_1(S'') + h_2(S''), 1\right).$$

Hence, $\underline{b}(\underline{x}) = \underline{a}\left(\underline{x}h_1(S'') + h_2(S'')\right)$. Since $S'' \in GL_{n+1}(\mathbb{F}_q)$, $h_1(S'') \in GL_n(\mathbb{F}_q)$ and it follows that $\left(h_1(S''), h_2(S'')\right)$ is an affine equivalence pair between $\underline{a}$ and $\underline{b}$, i.e. $(\underline{a}, \underline{b}) \in L_{dIP1S}$. □

Note that in this paper, we are interested in the finding of a solution of PLE (resp. IP1S) rather than deciding if such a solution exists. However, this result permits in fact to transform efficiently any algorithm dedicated to PLE to an algorithm for solving IP1S. Indeed, let the notations be as in the proof of proposition 1 and $(\underline{a}, \underline{b})$ be an instance of IP1S. Any linear equivalence S'' for $f(\underline{a}, \underline{b}) = \left(\underline{A}, \underline{B}\right)$ can be efficiently transformed into an affine equivalence pair $\left(h_1(S''), h_2(S'')\right)$ for $(\underline{a}, \underline{b})$. Thus, any solution given by a PLE algorithm, on input $\left(\underline{A}, \underline{B}\right)$, can be easily transformed to a solution for IP1S, i.e. an affine equivalence pair for $(\underline{a}, \underline{b})$.

On the other hand, we have the following (less surprising) result:

Proposition 2 *dPLE is polynomial-time many-one reducible to dIP1S.*

Proof. Let $(\underline{a},\underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$. For proving that dPLE$\leq_p^m$dIP1S, we define $f : \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u \to \mathbb{F}_q[\underline{x}]^{(D+1)\cdot u} \times \mathbb{F}_q[\underline{x}]^{(D+1)\cdot u}$ in the following way. For all $(\underline{a},\underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$, we have:

$$f\big(\underline{a},\underline{b}\big) = \big(\underline{A},\underline{B}\big),$$

with $\underline{A} = \big(\underline{a}^{(D)},\underline{a}^{(D-1)},\ldots,\underline{a}^{(0)}\big)$ and $\underline{B} = \big(\underline{b}^{(D)},\underline{b}^{(D-1)},\ldots,\underline{b}^{(0)}\big)$.
Let $(\underline{a},\underline{b}) \in L_{dPLE}$, i.e. $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, for some $S \in GL_n(\mathbb{F}_q)$.
According to property 2, we have $\underline{b}^{(d)}(\underline{x}) = \underline{a}^{(d)}(\underline{x}S)$, for all $d, 0 \leq d \leq D$. Thus $\underline{B}(\underline{x}) = \underline{A}(\underline{x}S)$, and $(S,\underline{0_n})$ is an affine equivalence pair between $\underline{A}$ and $\underline{B}$, i.e. $f(\underline{a},\underline{b}) = (\underline{A},\underline{B}) \in L_{dIP1S}$.

Now let $f\big(\underline{a},\underline{b}\big) = \big(\underline{A},\underline{B}\big) \in L_{dIP1S}$, i.e. $\underline{B}(\underline{x}) = \underline{A}(\underline{x}S' + \underline{T}')$, for some $(S',\underline{T}') \in AGL_n(\mathbb{F}_q)$. By the very construction of f, $\underline{B}(\underline{x}) = \underline{A}(\underline{x}S' + \underline{T}')$ implies that $\underline{b}^{(d)}(\underline{x}) = \underline{a}^{(d)}(\underline{x}S' + \underline{T}')$, for all $d, 0 \leq d \leq D$. We then have according to property 1 that:

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

By property 2, we get that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S')$, i.e. S' is a linear equivalence matrix between $\underline{a}$ and $\underline{b}$, proving that $(\underline{a},\underline{b}) \in L_{dPLE}$. □

Let the notations be as in the proof of proposition 2 and $(\underline{a},\underline{b})$ be an instance of PLE. If $(S,\underline{T})$ is an affine equivalence pair, between $f(\underline{a},\underline{b}) = \big(\underline{A},\underline{B}\big)$, then S is a linear equivalence matrix between $\big(\underline{A},\underline{B}\big)$, and thus between $(\underline{a},\underline{b})$. Thus, from any solution given by an IP1S algorithm, on input $\big(\underline{A},\underline{B}\big)$, one can easily construct a solution to PLE for $(\underline{a},\underline{b})$.
Finally, from propositions 1 and 2, we deduce:

Corollary 1 *$dPLE \equiv_p^m dIP1S$.*

This equivalence result also holds for PLE and IP1S (the search problems associated to dPLE and dIP1S). Indeed, aboves proofs construct a solution of PLE (resp. IP1S) from one of IP1S (resp. PLE). Thus, we can w.l.o.g restrict our attention to only one of these problems. Hereafter, we will focus on PLE. We have chosen more particularly this problem since it seems to have more useful algorithmic properties.

4 Properties of PLE

We present in this part new properties of PLE. In 4.1, we give a strong relation between the Jacobian matrices of an instance $(\underline{a},\underline{b})$ of PLE and solutions of this instance. In 4.2, we show that structural properties of PLE permit to obtain linear equations in the components of a linear equivalence matrix (provided such a matrix exists).

4.1 Differential Properties

In the one variable case (i.e. $n = 1$), PLE can be reformulated as follows: given polynomials $a_1(x), \ldots, a_u(x)$ and $b_1(x), \ldots, b_u(x)$ in $\mathbb{F}_q[x]$, find - if any - $s \in \mathbb{F}_q$, such that the equality $b_i(x) = a_i(xs)$ holds for all $i, 1 \leq i \leq u$. When computing the formal derivatives of these equalities, we get that s must be such that:

$$\frac{db_i}{dx}(x) = s\frac{da_i}{dx}(xs), \text{ for all } i, 1 \leq i \leq u.$$

Thus, if $\frac{da_i}{dx}(0) \neq 0$, for some i, then $s = \frac{\frac{db_i}{dx}(0)}{\frac{da_i}{dx}(0)}$. The next theorem, which is is the main result of this section, extend this idea to multivariate polynomials.

Theorem 1 *If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, for some $S \in GL_n(\mathbb{F}_q)$, then:*

$$J_{\underline{b}}(\underline{x}) = J_{\underline{a}}(\underline{x}S)S^t,$$

$J_{\underline{a}}(\underline{x}S) = \left\{\frac{\partial a_i}{\partial x_j}(\underline{x}S)\right\}_{1 \leq j \leq n}^{1 \leq i \leq u}$ and $J_{\underline{b}}(\underline{x}) = \left\{\frac{\partial b_i}{\partial x_j}(\underline{x})\right\}_{1 \leq j \leq n}^{1 \leq i \leq u}$ being the Jacobian matrices of $\underline{a}$ evaluated in $\underline{x}S$ and of $\underline{b}$ evaluated in $\underline{x}$, respectively.

¿From this theorem, we deduce the following corollaries:

Corollary 2 *Let $S \in GL_n(\mathbb{F}_q)$ be such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, and $(\underline{p}', \underline{p}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$ be such that $\underline{p}' = \underline{p}S$. Then:*

$$\begin{aligned} &i)\ J_{\underline{b}}(\underline{p}) = J_{\underline{a}}(\underline{p}')S^t \\ &ii)\ Ker\big(J_{\underline{a}}^t(\underline{p}')\big) = Ker\big(J_{\underline{b}}^t(\underline{p})\big)S \end{aligned}$$

Proof. $i)$ is obvious since $\underline{p}' = \underline{p}S$.
For $ii)$, let $\underline{k_a} \in Ker\big(J_{\underline{a}}^t(\underline{p}')\big)$, we have $\underline{k_a}S^{-1}J_{\underline{b}}^t(\underline{p}) = \underline{k_a}J_{\underline{a}}^t(\underline{p}') = \underline{0_u}$, therefore $\underline{k_a}S^{-1} \in Ker\big(J_{\underline{b}}^t(\underline{p})\big)$, i.e. $\underline{k_a} \in Ker\big(J_{\underline{b}}^t(\underline{p})\big)S$.
Now, let $\underline{k}' = \underline{k_b}S \in Ker\big(J_{\underline{b}}^t(\underline{p})\big)S$, we have $\underline{0_u} = \underline{k_b}J_{\underline{b}}^t(\underline{p}) = \underline{k}'J_{\underline{a}}^t(\underline{p}')$, i.e. $\underline{k}' \in Ker\big(J_{\underline{a}}^t(\underline{p}')\big)$. Thus, $Ker\big(J_{\underline{a}}^t(\underline{p}')\big) = Ker\big(J_{\underline{b}}^t(\underline{p})\big)S$. □

Corollary 3 *If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, for some $S \in GL_n(\mathbb{F}_q)$, then:*

$$J_{\underline{b}^{(d)}}(\underline{x}) = J_{\underline{a}^{(d)}}(\underline{x}S)S^t, \textit{for all } d, 0 \leq d \leq D.$$

$J_{\underline{a}^{(d)}}(\underline{x}S)$ and $J_{\underline{b}^{(d)}}(\underline{x})$ being the Jacobian matrices of $\underline{a}^{(d)}$ evaluated in $\underline{x}S$ and of $\underline{b}^{(d)}$ evaluated in $\underline{x}$, respectively.

4.2 Structural Properties

For each homogeneous polynomial $p \in \mathbb{F}_q[\underline{x}]$ of degree two there exists $Q \in \mathcal{M}_{n,n}(\mathbb{F}_q)$, such that $p(\underline{x}) = \underline{x}Q\underline{x}^t$. This matrix can be easily constructed from the knowledge of the coefficients of the terms of p, but is not unique in general. For fields of characteristic $\neq 2$, provided that Q is symmetric (resp. upper triangular, lower triangular) such a representation is unique. For fields of characteristic 2, the representation is unique if Q is upper triangular or lower triangular.

Corollary 4 *Let $Q_{a_i}, Q_{b_i} \in \mathcal{M}_{n,n}(\mathbb{F}_q)$ be, for all $i, 1 \leq i \leq u$, the unique matrices*[1] *such that $a_i^{(2)}(\underline{x}) = \underline{x}Q_{a_i}\underline{x}^t$ and $b_i^{(2)}(\underline{x}) = \underline{x}Q_{b_i}\underline{x}^t$. If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, for some $S \in GL_n(\mathbb{F}_q)$, then:*

$$i)\, Q_{b_i} = SQ_{a_i}S^t, \text{ for all } i, 1 \leq i \leq u$$
$$ii)\, Ker(Q_{a_i}) = Ker(Q_{b_i})S, \text{ for all } i, 1 \leq i \leq u.$$

Proof. For $i)$, we obtain by property 2 that if $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, for some $S \in GL_n(\mathbb{F}_q)$, then $\underline{b}^{(2)}(\underline{x}) = \underline{a}^{(2)}(\underline{x}S)$. Thus, for all $i, 1 \leq i \leq u$, we have $\underline{x}Q_{b_i}\underline{x}^t = \underline{x}SQ_{a_i}S^t\underline{x}^t$, i.e. $Q_{b_i} = SQ_{a_i}S^t$.
For $ii)$, let $\underline{k_{a_i}} \in Ker(Q_{a_i})$, we have $\underline{k_{a_i}}S^{-1}Q_{b_i} = \underline{k_{a_i}}Q_{a_i}S^t = \underline{0_n}S^t = \underline{0_n}$, thus $\underline{k_{a_i}}S^{-1} \in Ker(Q_{b_i})$, i.e. $\underline{k_{a_i}} \in Ker(Q_{b_i})S$, for all $i, 1 \leq i \leq u$.
Now, let $\underline{k'} = \underline{k_{b_i}}S \in Ker(Q_{b_i})S$, we have $\underline{0_n} = \underline{k_{b_i}}Q_{b_i}(S^t)^{-1} = \underline{k'}Q_{a_i}$, and thus $\underline{k'} \in Ker(Q_{a_i})$, for all $i, 1 \leq i \leq u$. □

We finish this part by extending, thanks to property 2, a result given in [2].

Corollary 5 *Let $Q_{a_i}, Q_{b_i} \in \mathcal{M}_{n,n}(\mathbb{F}_q)$ be, for all $i, 1 \leq i \leq u$, the unique matrices such that $a_i^{(2)}(\underline{x}) = \underline{x}Q_{a_i}\underline{x}^t$ and $b_i^{(2)}(\underline{x}) = \underline{x}Q_{b_i}\underline{x}^t$. Moreover, let $S \in GL_n(\mathbb{F}_q)$ be such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$. If there exists $j, 1 \leq j \leq n$, such that Q_{b_j} is invertible then for all $i, 1 \leq i \neq j \leq n$:*

$$S^tQ_{b_j}^{-1}Q_{b_i} = Q_{a_j}^{-1}Q_{a_i}S^t. \tag{6}$$

Proof. According to corollary 4, we have $Q_{b_i} = SQ_{a_i}S^t$, for all $i, 1 \leq i \leq u$. Moreover, since Q_{b_j} and S are invertible, we get that $S^{-1} = Q_{a_j}S^tQ_{b_j}^{-1}$. It follows that, for all $i, 1 \leq i \neq j \leq n$, $Q_{a_j}S^tQ_{b_j}^{-1}Q_{b_i} = Q_{a_i}S^t$. Finally, since Q_{b_j} is invertible then Q_{a_j} is also invertible and we get that $S^tQ_{b_j}^{-1}Q_{b_i} = Q_{a_j}^{-1}Q_{a_i}S^t$, for all $i, 1 \leq i \neq j \leq n$. □

We stress that this corollary extends the result given in [2], since equation (6) holds for all instances of PLE whereas the result quoted in [2] holds for instances of PLE composed of homogeneous polynomials of degree 2 only.

5 The PLE algorithm

Levy-dit-Vehel and Perret have linked PLE with the problem of finding common zeroes of multivariate polynomials [7]. We go one step further in this section. Indeed, we show that a partial knowledge of a linear equivalence matrix allows us to recover it entirely by solving a suitable linear system of equations.

---

[1] in upper triangular form, lower triangular form, or symmetric form, if such a matrix exists

5.1 Description of the PLE algorithm

In the sequel, we always suppose that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, for some $S \in GL_n(\mathbb{F}_q)$.
Let us present now the main ideas of our algorithm.

How to easily recover linear equations in the components of S?

We describe here how to obtain, from properties described in section 4, linear equations in the components of S. Indeed, let Q_{a_i} and Q_{b_i} be, for all $i, 1 \leq i \leq u$, defined as in corollary 4. By corollary 5, we have that, whenever Q_{b_j} is invertible for some $j, 1 \leq j \leq n$, then $S^t Q_{b_j}^{-1} Q_{b_i} = Q_{a_j}^{-1} Q_{a_i} S^t$, for all $i, 1 \leq i \neq j \leq n$. Moreover, according to corollary 2, we have additionally that $J_{\underline{b}}(\underline{0_n}) = J_{\underline{a}}(\underline{0_n}) S^t$. Thus, S is a particular solution of the following linear system of equations, with unknowns the components of $X \in \mathcal{M}_{n,n}(\mathbb{F}_q)$:

$$\begin{cases} J_{\underline{b}}(\underline{0_n}) = J_{\underline{a}}(\underline{0_n}) X^t \\ X^t Q_{b_j}^{-1} Q_{b_i} = Q_{a_j}^{-1} Q_{a_i} X^t, \forall i, j 1 \leq i \neq j \leq n, \text{ s.t. } Q_{b_j} \text{ is invertible} \end{cases} \tag{7}$$

How to start the algorithm?

In our algorithm, we need to find pairs $(\underline{p'}, \underline{p}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$, such that $\underline{p'} = \underline{p}S$. Such a pair can obviously be recovered by randomly selecting $\underline{p} \in \mathbb{F}_q^n$ and then performing an exhaustive search, over $\mathbb{F}_q^n$, to find the corresponding vector $\underline{p'} = \underline{p}S$. In many cases, we can, thanks to properties of section 4, significantly decrease the cost of this exhaustive search.
Indeed, according to corollary 2, $Ker\big(J_{\underline{a}}^t(\underline{0_n})\big) = Ker\big(J_{\underline{b}}^t(\underline{0_n})\big)S$. Consequently, any vector $\underline{p} \in Ker\big(J_{\underline{b}}^t(\underline{0_n})\big)$ is mapped to $Ker\big(J_{\underline{a}}^t(\underline{0_n})\big)$, i.e. there exists $\underline{p'} \in Ker\big(J_{\underline{a}}^t(\underline{0_n})\big)$ such that $\underline{p'} = \underline{p}S$. Thus, if we chose a vector $\underline{p} \in Ker\big(J_{\underline{b}}^t(\underline{0_n})\big)$ then $\underline{p'} = \underline{p}S$ can be recovered by listing all elements of $Ker\big(J_{\underline{a}}^t(\underline{0_n})\big)$, rather than all $\mathbb{F}_q^n$.
Similarly, using the quadratic parts of the polynomials of $\underline{a}$ and $\underline{b}$, we obtain, according to corollary 4, that for all $i, 1 \leq i \leq u$, any vector $\underline{p} \in Ker(Q_{b_i})$ is mapped to an element of $Ker(Q_{a_i})$. Thus, by choosing $\underline{p} \in Ker(Q_{b_i})$, we can recover $\underline{p'} = \underline{p}S$ by performing an exhaustive search over $Ker(Q_{a_i})$.

How to use Jacobian matrices?

Let $(\underline{p'}, \underline{p}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$ be such that $\underline{p'} = \underline{p}S$. According to corollary 2, we have $J_{\underline{b}}(\underline{p}) = J_{\underline{a}}(\underline{p'}) S^t$. From this equality, we obtain $n \cdot u$ linear equations in n^2 unknowns (the components of S), $n \cdot Rank\big(J_{\underline{a}}(\underline{p'})\big)$ of which are linearly independent. When $Rank\big(J_{\underline{a}}(\underline{p'})\big) < n$, all the solutions found do not necessarily give a linear equivalence matrix between $\underline{a}$ and $\underline{b}$. To eliminate superfluous solutions, we need to find new linear equations in the components of S. To do so, we increase the number of pairs $(\underline{p'}, \underline{p}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$, such that $\underline{p'} = \underline{p}S$. When one has found $P = \{(\underline{p'_j}, \underline{p_j})_{1 \leq j \leq \ell}\}$, such that $\underline{p'_j} = \underline{p_j}S$, for all $j, 1 \leq j \leq \ell$, then S is a solution of the following linear system of equations:

$$\begin{cases} J_{\underline{b}}(\underline{p_j}) = J_{\underline{a}}(\underline{p'_j}) X^t, \text{ for all } j, 1 \leq j \leq \ell. \\ \underline{p'_j} = \underline{p_j} X, \text{ for all } j, 1 \leq j \leq \ell. \end{cases}$$

In other words, $n \cdot \ell$ linear equations, given by P, relating the components of S are transformed into $n \cdot \ell \cdot (u+1)$ linear equations in the components of S. Thus ℓ must be chosen such that $n \cdot \ell \cdot (u+1) = n^2$, i.e. $\ell \approx \left\lceil \frac{n}{u+1} \right\rceil$ in order to obtain in this way (and without using (7)), n^2 linear equations in the components of S. However, we point out that equations generated in this way are not necessarily linearly independent.
Finally, we can also use a structural property of PLE to decrease the minimal value of ℓ required. Indeed, according to corollary 3, we have for all $d, 1 \leq d \leq D$:

$$J_{\underline{b}^{(d)}}(\underline{p}) = J_{\underline{a}^{(d)}}(\underline{p}S)S^t, \text{ for all } \underline{p} \in \mathbb{F}_q^n.$$

Notice that this last equation also holds for $d = 0$, but does not permit to get linear equations. Therefore, if $P = \{(\underline{p'_j}, \underline{p_j})_{1 \leq j \leq \ell}\}$ is a set of vector such that $\underline{p'_j} = \underline{p_j}S$, for all $j, 1 \leq j \leq \ell$, then S is a solution of the following linear system of equations:

$$\begin{cases} J_{\underline{b}}(\underline{p_j}) = J_{\underline{a}}(\underline{p'_j})X^t, \text{ for all } j, 1 \leq j \leq \ell. \\ J_{\underline{b}^{(d)}}(\underline{p_j}) = J_{\underline{a}^{(d)}}(\underline{p'_j})S^t, \text{ for all } d, 1 \leq d \leq D \text{ and for all } j, 1 \leq j \leq \ell. \\ \underline{p'_j} = \underline{p_j}X, \text{ for all } j, 1 \leq j \leq \ell. \end{cases} \quad (8)$$

In the sequel, $Sys(P)$ shall denote the linear system of equations obtained from (7) and (8).
We are now ready to present the PLE algorithm.

The algorithm
For a given $\ell \geq 1$, we select ℓ distinct (non-zero) vectors $\underline{p_1}, \ldots, \underline{p_\ell}$ and perform a so-called selective exhaustive search, which is detailed after the description of the PLE algorithm, to recover the corresponding vectors $\underline{p'_1} = \underline{p_1}S, \ldots, \underline{p'_\ell} = \underline{p_\ell}S$. The aim of this selective exhaustive search is to minimize the cost of constructing a set $P = \{(\underline{p'_j}, \underline{p_j})_{1 \leq j \leq \ell}\}$ such that $\underline{p'_j} = \underline{p_j}S$, for all $j, 1 \leq j \leq \ell$. We then compute the solutions of $Sys\big((\underline{p'_j}, \underline{p_j})_{1 \leq j \leq \ell}\big)$, denoted by $Sol\big(Sys\big((\underline{p'_j}, \underline{p_j})_{1 \leq j \leq \ell}\big)\big)$ in our algorithm, and the number of solutions of this linear system of equations. If it has less than C solutions (C is a small constant given in input of the algorithm), we try to find a solution of this system which is at the same time a linear equivalence matrix. If such a matrix exists then we return it. Otherwise and if, after having tried all the possible vectors $\underline{p'_1}, \ldots, \underline{p'_\ell}$ corresponding to $\underline{p_1}, \ldots, \underline{p_\ell}$, we have not obtained a linear equivalence matrix, we increment ℓ by 1 and restart the PLE algorithm with this new value of ℓ.
In the PLE algorithm, we use an auxiliary function, which we call Order, taking as input $n+1$ pairs of sets of vectors and returning these sets sorted in decreasing order (with respect to the number of elements in these sets).

The PLE algorithm
Input: $(\underline{a},\underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$, $(\ell, C) \in \mathbb{N}^* \times \mathbb{N}^*$.
Output: $S \in GL_n(\mathbb{F}_q)$, such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$.
$Sol_0 \leftarrow Sol\big(Sys(\underline{0_n}, \underline{0_n})\big)$
If $|Sol_0| \leq C$ **then**
    If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, for some $S \in Sol_0$ **then** return S
EndIf
`Selective Exhaustive Search: towards finding suitable pairs` $(\underline{p'}, \underline{p})$
$Aux \leftarrow \Big(\big(Ker\big(J^t_{\underline{a}}(\underline{0_n})\big), Ker\big(J^t_{\underline{b}}(\underline{0_n})\big)\big), (Ker(Q_{a_1}), Ker(Q_{b_1})), \ldots, (Ker(Q_{a_n}),$
    $Ker(Q_{b_n}))\Big)$
$\big((A_0, B_0), \ldots, (A_n, B_n)\big) \leftarrow \text{Order}\big(Aux\big)$ and $(A_{n+1}, B_{n+1}) \leftarrow (\mathbb{F}_q^n, \mathbb{F}_q^n)$
Let k be the minimum index such that $|\cup_{j=0}^{k} A_j| \geq \ell$
For i **from** 1 **to** k **do**
    $B_i \leftarrow B_i \setminus \cup_{j=0}^{i-1} B_j$ and $A_i \leftarrow A_i \setminus \cup_{j=0}^{i-1} A_j$
EndFor
$k' \leftarrow \ell - \sum_{j=0}^{k-1} |A_i|$ and $Aux \leftarrow A_0^{|A_0|} \times A_1^{|A_1|} \times \cdots \times A_k^{k'}$
`Select` $|B_0|$ `vectors in` B_0`,` $|B_1|$ `vectors in` B_1`, ..., and` k' `vectors in` B_k
Randomly choose $(p_1, \ldots, p_\ell) \in B_0^{|B_0|} \times B_1^{|B_1|} \times \cdots \times B_k^{k'}$
`Search of a linear equivalence matrix`
While $\underline{b}(\underline{x}) \neq \underline{a}(\underline{x}S)$ **or** $Aux \neq \emptyset$ **do**
    `Select` $|A_0|$ `vectors in` A_0`,` $|A_1|$ `vectors in` A_1`,..., and` k' `vectors in` A_k
    Randomly choose $(p'_1, \ldots, p'_\ell) \in A_0^{|A_0|} \times A_1^{|A_1|} \times \cdots \times A_k^{k'}$
    $P \leftarrow \{(\underline{p'_j}, \underline{p_j})_{1 \leq j \leq \ell}\}$ and $Aux \leftarrow Aux \setminus \{p'_1, \ldots, p'_\ell\}$
    $Sol_P \leftarrow Sol\big(Sys(P)\big)$
    If $|Sol_P \cap Sol_0| \leq C$ **then**
        If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, for some $S \in Sol_P \cap Sol_0$ **then** return S
    EndIf
EndWhile

The Selective Exhaustive Search

Let the notations be as in the PLE algorithm. One can see at once that, for all $i, 0 \leq i \leq n+1$, we have $A_i = B_i S$. We stress that such a property also holds after the first for loop. Moreover at each iteration of the PLE algorithm, by the very definition of k, it holds that $|\cup_{j=0}^{k-1} A_j| < \ell$, and thus $k' > 0$. Moreover, we have that $\ell = k' + \sum_{j=0}^{k-1} |B_i| = k' + \sum_{j=0}^{k-1} |A_i|$, since $|A_i| = |B_i|$, for all $i, 0 \leq i \leq n+1$.

In order to recover ℓ pairs of vectors $(\underline{p'_j}, \underline{p_j})_{1 \leq j \leq \ell}$, such that $\underline{p'_j} = \underline{p_j} S$, for all $j, 1 \leq j \leq \ell$, we select $|B_0|$ vectors $\underline{p_1}, \ldots, \underline{p}_{|B_0|} \in B_0$, and perform an exhaustive search over $A_0(= B_0 S)$ to recover the corresponding vectors $\underline{p'_1} = \underline{p_1} S, \ldots,$ $\underline{p'}_{|B_0|} = \underline{p}_{|B_0|} S$. We complete these $|B_0|$ vectors by choosing $|B_1|$ new vectors $\underline{p}_{|B_0|+1}, \ldots, \underline{p}_{|B_0|+|B_1|} \in B_1$. The corresponding vectors $\underline{p'}_{|B_0|+1} = \underline{p}_{|B_0|+1} S, \ldots,$ $\underline{p'}_{|B_0|+|B_1|} = \underline{p}_{|B_0|+B_1|} S$ are recovered by performing an exhaustive search over $A_1(= B_1 S)$. Finally, we complete the $\sum_{j=0}^{k-1} |B_i|$ vectors already chosen by selecting $k'(=\ell-\sum_{j=0}^{k-1}|B_i|)$ new vectors $\underline{p}_{\ell-k'},\ldots,\underline{p_\ell}\in B_k$. The corresponding vectors $\underline{p'}_{\ell-k'}=\underline{p}_{\ell-k'}S,\ldots,\underline{p'_\ell}=\underline{p_\ell}S$ are recovered by performing an exhaustive search over $A_k(=B_kS)$. Since, by construction, $|A_0|\leq|A_1|\leq\cdots\leq|A_k|$, we minimize in this way the cost of an exhaustive search for recovering the vectors $\underline{p'_1}=\underline{p_1}S,\ldots,\underline{p'_\ell}=\underline{p_\ell}S$.

5.2 Complexity

Let $\ell^*\in\mathbb{N}$ be the minimum value for which PLE returns a solution, i.e. the minimum number of pairs in P, for which $Sys(P)$ has n^2 linearly independent equations. As explained in 5.1, $\underline{b}(\underline{x})=\underline{a}(\underline{x}S)$, for some $S\in GL_n(\mathbb{F}_q)$, implies that the linear equivalence matrix S verifies the following linear equations:

$$\begin{cases} J_{\underline{b}}(\underline{0_n})=J_{\underline{a}}(\underline{0_n})S^t \\ S^tQ_{b_j}^{-1}Q_{b_i}=Q_{a_j}^{-1}Q_{a_i}S^t, \forall 1\leq j\leq n, \text{ s.t. } Q_{b_j} \text{ is invertible and } \forall 1\leq i\neq j\leq n \end{cases}$$

These equalities allow us to obtain say nb_0 linearly independent equations in the components of S. We would like to emphasize that these equations are obtained in polynomial-time. Thus, if $nb_0=n^2$ then our algorithm recovers S in polynomial-time.
Otherwise, if $nb_0<n^2$, we have to find $\ell^*\geq 1$ pairs of non-zero vectors $(\underline{p},\underline{p'})$, such that $\underline{p'}=\underline{p}S$. The cost of recovering these ℓ^* additional pairs being bounded from above by $q^{n\ell^*}$, the complexity of the PLE algorithm is:

$$O(n^6q^{n\ell^*}),$$

which is the cost of solving a linear system of n^2 unknowns times the cost of recovering ℓ^* suitable pairs of vectors.
It seems difficult to obtain a precise value of ℓ^*. Anyway, in practice it appears that it is on the order of $\left\lceil\frac{n}{u+1}\right\rceil$. Finally, we mention that in order to minimize the number of pairs ℓ^* which has to be recovered, we can exploit a powerful idea that we shall call *exponentiation process*. It will be described in an extended version of this paper.

5.3 Practical behaviour

We conclude this paper by giving some experimental results obtained with the PLE algorithm. The instances $(\underline{a},\underline{b})$ of PLE have been generated in the following way. The polynomials of $\underline{a}$ have been randomly chosen of degree 2 or 3 (or more precisely with terms of total degree at most 2 or 3). To construct the polynomials of $\underline{b}$, we have randomly chosen $S\in GL_n(\mathbb{F}_q)$ and computed $\underline{b}(\underline{x})=\big(a_1(\underline{x}S),\cdots,a_u(\underline{x}S)\big)$. The PLE algorithm described in 5.1 has been implemented using Magma software [8]. We have chosen the constant C (given in input of the PLE algorithm) equals to 10000. The results, obtained on a standard PC, are quoted in the following table. We mention that the times given in this table are in fact average times, obtained with our algorithm, for solving 10 instances of PLE (with u, n and q given).

n	u	q	degree	Time	degree	Time
50	50	$\mathbb{F}_{257}$	2	≈ 0.3 s.	3	≈ 10 s.
50	45	$\mathbb{F}_{257}$	2	≈ 10 min.	3	≈ 6 h.
60	60	$\mathbb{F}_{11}$	2	≈ 0.2 s.	3	≈ 10 s.
60	55	$\mathbb{F}_{11}$	2	≈ 2 min.	3	≈ 1 h.
60	50	$\mathbb{F}_{11}$	2	≈ 2 min.	3	≈ 1 h.
70	70	$\mathbb{F}_2$	2	≈ 10 s.	3	≈ 5 min.
70	65	$\mathbb{F}_2$	2	≈ 10 s.	3	≈ 5 min.
70	60	$\mathbb{F}_2$	2	≈ 9 s.	3	≈ 5 min.
70	55	$\mathbb{F}_2$	2	≈ 9 s.	3	≈ 5 min.

We would like to emphasize that the algorithms described in [7] have also been tested on these instances. The results are not quoted since these algorithms do not terminate (in a reasonable time). Anyway, we mention that in [11], the algorithms of [7] have been compared with a restricted version of the PLE algorithm described here. These experiments have been done on smaller instances of PLE (in terms of u, n and q) than the ones quoted in the above table. It appears that the PLE algorithm is much more efficient than the algorithms of [7] (which perform better than the algorithm described in [3]). This is mainly due to the fact that we have replaced the computation of Gröbner Bases by a Gaussian elimination.

Interpretation of the results

We first mention that the case $u \approx n$ is the most interesting for cryptographic applications of PLE. Indeed, in this setting it is very likely that an instance admits a unique solution (*see* [7] for further details). Moreover, $J_{\underline{b}}(\underline{0_n}) = J_{\underline{a}}(\underline{0_n})S^t$, allows us to obtain $n * Rank(J_{\underline{a}}(\underline{e_0}))$ linearly independent equations in the components S. Since $u \approx n$, then $Rank(J_{\underline{a}}(\underline{0_n}))$ is also close to n. Therefore, even if S is not uniquely determined by these equations, it is then very likely that a very little partial knowledge of S allows us to obtain a linear system of equation with less than C solutions. Since C is very small, we can quickly find if one of these C solutions is at the same time a linear equivalence matrix. Typically, in our experiments it has been sufficient to recover one pair $(\underline{p'}, \underline{p})$ such that $\underline{p'} = \underline{p}S$, confirming, at least for these parameters, that ℓ^* is close to $\left\lceil \frac{n}{u+1} \right\rceil$. Note that this pair is recovered efficiently using our selective exhaustive search.

Let us now analyze our results.

When $u = n$, and for $p = 257$ (resp. $p = 11$) the matrix $J_{\underline{a}}(\underline{0_n})$ was always invertible (in the ten instances generated). In this case, the solution is simply obtained by computing the transpose of $J_{\underline{a}}(\underline{0_n})^{-1} J_{\underline{b}}(\underline{0_n})$. For $p = 2$, it was not the case and we had to find only one pair $(\underline{p'}, \underline{p})$ such that $\underline{p'} = \underline{p}S$ in order to solve PLE. For this reason, our algorithm for $u = n$ is faster for $p = 257$ (resp. $p = 11$) than for $p = 2$. This result is in fact not surprising since the probability that a matrix $M \in \mathcal{M}_{n,n}(\mathbb{F}_q)$ is invertible is larger in $\mathbb{F}_{257}$ (resp. $\mathbb{F}_{11}$) than in

$\mathbb{F}_2$. For instances $(\underline{a}, \underline{b})$ of PLE of degree 2, we can efficiently check if S is indeed a linear equivalence matrix between $(\underline{a}, \underline{b})$. Let $A, B \in \mathcal{M}_{n,u}(\mathbb{F}_q)$ such that $\underline{a}^{(1)}(\underline{x}) = \underline{x}A$ and $\underline{b}^{(1)}(\underline{x}) = \underline{x}B$. Moreover, let Q_{a_i}, Q_{b_i} be, for all $i, 1 \leq i \leq u$, the unique matrices such that $a_i^{(2)}(\underline{x}) = \underline{x}Q_{a_i}\underline{x}^t$ and $b_i^{(2)}(\underline{x}) = \underline{x}Q_{b_i}\underline{x}^t$. According to property 2, we have $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$ iff $B = SA$ and $Q_{b_i} = SQ_{a_i}S^t$, for all $i, 1 \leq i \leq u$. Therefore, to check whether $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, we just have to compute product of matrices and compare these matrices. For an instance $(\underline{a}, \underline{b})$ of degree 3, such a manipulation is possible only for the homogeneous components of degree 1, and 2. But, in order to check whether $\underline{b}^{(3)}(\underline{x}) = \underline{a}^{(3)}(\underline{x}S)$ or not, we have to compute formally the polynomials $\underline{a}^{(3)}(\underline{x}S)$, which is much more costly than computing product of matrices (explaining the significant difference of results between instances of degree 2 and 3).

6 Conclusion

We have proved in this paper that IP1S and PLE are equivalent. Moreover, using a differential approach of PLE, we have presented a fast algorithm for solving PLE (and consequently also IP1S). It appears that, with the parameters proposed in [9], schemes based on IP1S are far from achieving the security level required for cryptographic applications. We recall that, initially, the security level of schemes based on IP1S has been estimated to $q^{\sqrt{2}n^{3/2}}$[10].

7 Acknowledgements

I would like to thank F.Levy-dit-Vehel for helpful discussions related to this paper.

References

1. N. Courtois, L. Goubin, and J. Patarin: Improved Algorithms for Isomorphism of Polynomials. *Advances in Cryptology - EUROCRYPT '98, Lecture Notes in Computer Science,* vol. 1403, Springer-Verlag, pp. 84–200, 1998.
2. N. Courtois, L. Goubin, and J. Patarin: Improved Algorithms for Isomorphism of Polynomials - Extended Version. Available from `www.minrank.org`.
3. W.Geiselmann, W.Meier, and R.Steinwandt: An Attack on the Isomorphisms of Polynomials Problem with One Secret. Int. Journal of Information Security, Vol. 2(1): pp. 59–64, 2003.
4. O. Goldreich, S. Micali, and A. Wigderson: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM, Vol. 38(3) pp. 690–728, 1991.
5. M. R. Garey, and D. B. Johnson: Computers and Intractability. A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
6. S. Goldwasser, S. Micali, and C. Rackoff: The Knowledge Complexity of Interactive Proof Systems. SIAM J. on Computing, Vol. 18, pp. 186–208, 1989.

7. F. Levy-dit-Vehel, and L. Perret: Polynomial equivalence problems and applications to multivariate cryptosystems. *Progress in Cryptology - INDOCRYPT 2003, Lecture Notes in Computer Science,* vol. 2904, pp. 235–251, 2003.
8. http://magma.maths.usyd.edu.au/magma/
9. J. Patarin: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms. *Advances in Cryptology - EUROCRYPT '96, Lecture Notes in Computer Science,* vol. 1070, Springer-Verlag, pp. 33–48, 1996.
10. J. Patarin:Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms - Extended Version. Available from `www.minrank.org/hfe/`.
11. L. Perret, and A. Bayad: A differential approach to a polynomial equivalence problem, in Proceedings of ISIT 2004, extended abstract, pp. 142, 2004.