Polynomial equivalence problems and applications to multivariate cryptosystems

Françoise Levy-Dit-Vehel, Ludovic Perret

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Polynomial equivalence problems and applications to multivariate cryptosystems

Françoise Levy-dit-Vehel — Ludovic Perret

N° 5119

Février 2004

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--5119--FR+ENG

INRIA
ROCQUENCOURT

Polynomial equivalence problems and applications to multivariate cryptosystems

Françoise Levy-dit-Vehel * , Ludovic Perret*

Thème 2 — Génie logiciel
et calcul symbolique
Projet Codes

Rapport de recherche n° 5119 — Février 2004 — 22 pages

Abstract: At Eurocrypt'96, J.Patarin proposed a signature and authentication scheme whose security relies on the difficulty of the Isomorphism of Polynomials problem [P]. In this paper, we study a variant of this problem, namely the Isomorphism of Polynomials with one secret problem and we propose new algorithms to solve it, which improve on all the previously known algorithms. As a consequence, we prove that, when the number of polynomials (u) is close to the number of variables (n), the instances considered in [P] and [P1] can be broken. We point out that the case $n - u$ small is the most relevant one for cryptographic applications. Besides, we show that a large class of instances that have been presumed difficult in [P] and [P1] can be solved in deterministic polynomial time. We also give numerical results to illustrate our methods.

Key-words: multivariate polynomial equations, Isomorphism of Polynomials, Gröbner Bases.

* ENSTA, Laboratoire de Mathématiques appliquées, 32 bd Victor, 75739 Paris cedex 15.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Problèmes d'équivalence polynômiale et applications aux cryptosystèmes multivariés

Résumé : Le problème de l'isomorphisme de polynômes comme base à la construction de cryptosystèmes à clef publique (en particulier de schémas d'authentification et de signature) a été proposé par J. Patarin lors de la conférence Eurocrypt'96 [P]. Ici, nous étudions une variante de ce problème, connue sous le nom d'isomorphismes de polynômes à un secret, et nous proposons de nouveaux algorithmes pour le résoudre, qui améliorent tous les algorithmes antérieurs. Ceux-ci nous permettent de prouver que, lorsque le nombre de polynômes (u) est proche du nombre de variables (n), les instances considérées dans [P] et [P1] peuvent être cassées. Il est à préciser que le cas où $n - u$ est petit est le plus intéressant pour des applications cryptographiques. En outre, nous montrons qu'une classe importante d'instances présumées difficiles dans [P] et [P1] peut être résolue en temps polynômial. Nous terminons par des résultats numériques illustrant les performances de nos algorithmes.

Mots-clés : Equations polynômiales multivariées, Isomorphismes de polynômes, Bases de Gröbner.

1 Introduction

Alternatively to public key cryptosystems based on integer factorization and discrete log problems, there exists cryptographic schemes whose security relies on the difficulty of finding a common zero of a set of non linear polynomials in several variables. This problem is known to be solvable by Gröbner bases calculations[1]. Up to now, apart from the cryptanalysis of HFE by J.C Faugère *et al.* [FJ], this tool has not been used to attack these systems. The main reason is probably that there was no really efficient method to compute them. These past years, significant progress has been made [Fa99],[Fa02], which carried out to the design of a new efficient software to compute Gröbner bases: *fgb*[2].

In this paper, we are interested in variants of the *Isomorphism of Polynomials* (IP) problem - as introduced by J. Patarin in [P]. Our idea is to link these variants to the above problem of finding zeroes of a system of polynomials. Our approach is not only of theoretical interest but also gives in some cases efficient methods to solve these IP variants, since we are able to solve instances that are used in cryptographic applications. The variants we consider are the *Isomorphism of Polynomials with one secret* (IP1S) problem, and its linear counterpart. IP1S can be outlined as follows: given two sets of multivariate polynomials $\mathcal{A} = \{a_1(x_1, \cdots, x_n), \cdots, a_u(x_1, \cdots, x_n)\}$ and $\mathcal{B} = \{b_1(x_1, \cdots, x_n), \cdots, b_u(x_1, \cdots, x_n)\}$ over a finite field $\mathbb{F}_q$, find - if any - an invertible matrix S and a vector T such that $b_i(x_1, \cdots, x_n) = a_i((x_1, \cdots, x_n)S + T)$ for all i, $1 \leq i \leq u$. The linear variant of IP1S is the one where we only look for a matrix S (i.e. T is the null vector of $\mathbb{F}_q^n$).

In [P], it has been shown how to derive a signature and an authentication scheme from IP1S. It is believed more difficult [CGP] than the IP problem itself and as evidence of its hardness, it is shown in [CGP] that the IP1S problem is at least as difficult as the Graph Isomorphism (GI) problem or in other words, a deterministic polynomial time algorithm solving the IP1S problem would also solve the GI problem (a result which has not been achieved for the IP problem).

In this paper, for reasons we explain in the next section, we rename the IP1S problem into the *Polynomial Affine Equivalence* (PAE) problem. We here study the PAE problem and its linear variant, which we call the *Polynomial Linear Equivalence* (PLE) problem. Apart from [GMS], no algorithm has been designed for these problems. We present here new algorithms for solving them, based on the link we exhibit between them and that of finding zeroes of a system of polynomials. When one of the two sets of polynomials $\mathcal{A}$ or $\mathcal{B}$ induces a bijective mapping, we propose an algorithm of complexity $\mathcal{O}((n+1)D^n)$, where D is the maximum degree of the polynomials involved for solving the PAE problem, and of complexity $\mathcal{O}(nD^n)$ for the PLE problem. For the general case, we present algorithms of complexity $\mathcal{O}(f(n)D^n + g(n))$, where $f(n) \leq 2n, \forall n$, and $g(n)$ depends on the cardinality of the varieties involved.

[1] Note that Gröbner bases provide not only a tool for finding a zero of a system of polynomials, but permits in fact to find all the zeroes.

[2] http://calfor.lip6.fr/ jcf/Software/Fgb/index.html

The paper is organized as follows. We begin in section 2 by introducing our notations. We also define more formally the PAE and PLE problems, which are the main concerns of this paper. In section 3, we present some new properties of these problems. This section is divided into two parts, the first one investigates structural properties, whereas the other presents a geometrical interpretation of these problems. In section 4, we present two new algorithms for solving the PLE problem which are based on the properties of section 3. We compare in this part our algorithms with the one proposed in [GMS]. Independently from these new algorithms, we exhibit instances of the PLE problem which are solvable in deterministic polynomial time. In section 5, we generalize the algorithms of section 4 to the affine case.
The last part is devoted to applications: we investigate the security of cryptosystems based on the PAE and PLE problems. We prove that when the number u of polynomials and the number n of variables are such that $n-u \geq 0$ is small, the parameter sizes of the instances considered in [P] and [P1] do not guarantee a reasonable level of security. We point out that this case is the most relevant one for cryptographic applications (indeed, when $n-u$ small, the considered systems of polynomials have only one common zero, or at worse very few zeroes). We also show that a large class of instances that have been presumed difficult in [P] and [P1] can be solved in deterministic polynomial time. We give evidences of the efficiency of the methods we propose by presenting experimental results of our algorithms on presumably intractable instances. Moreover, we present an efficient general total break ciphertext attack on any encryption system whose security relies on the difficulty of the PAE problem. This is the case for example for restricted (to IP1S) versions of C^* [MI], HFE [P] and TTM [TTM]. We end the paper by some comments on the IP problem.

2 Preliminaries

Throughout this paper we use the following notations. We denote by $\mathbb{F}_q$ a finite field with q elements, by X the vector $(x_1, \cdots, x_n)$, by $\mathbb{F}_q[X] = \mathbb{F}_q[x_1, \ldots, x_n]$ the polynomial ring in the indeterminates $x_1, \cdots, x_n$ over $\mathbb{F}_q$, by $\mathcal{M}_{m,n}(\mathbb{F}_q)$ the set of $m \times n$ matrices whose components lie in $\mathbb{F}_q$ and by $GL_n(\mathbb{F}_q)$ the invertible matrices in $\mathcal{M}_{n,n}(\mathbb{F}_q)$. For a subset $V \subset \mathbb{F}_q^n$, we shall denote by $Span(V)$ the $\mathbb{F}_q$-vector space generated by all the linear combinations of vectors of V and by $dim_{\mathbb{F}_q}(Span(V))$ its dimension.

A *term* is a product of a field element by a product of the variables $x_1, \cdots, x_n$. We shall define the *total degree* of a term $cx_1^{\mu_1} \cdots x_n^{\mu_n}, c \in \mathbb{F}_q$ and $(\mu_1, \cdots, \mu_n) \in \mathbb{N}^n$ by the sum $\sum_{i=1}^{n} \mu_i$. As usual, the *head term* of a polynomial $p \in \mathbb{F}_q[X]$ is the biggest term of the terms of p (with respect to some admissible ordering on the terms) and the *degree* of this polynomial is the total degree of its head term.

Let $\mathcal{F} = \{f_1, \cdots, f_s\}$ be a set of polynomials in $\mathbb{F}_q[X]^s$, we shall say that $\mathcal{F}$ is *bijective* if the function $X \mapsto (f_1(X), \cdots, f_s(X))$ is a bijection. We shall denote by $V_{\mathcal{F}} = \{(z_1, \cdots, z_n) \in \bar{\mathbb{F}}_q : f_i(z_1, \cdots, z_n) = 0, \forall 1 \leq i \leq s\}$, where $\bar{\mathbb{F}}_q$ is the algebraic closure of $\mathbb{F}_q$, the *variety* associated to the ideal $< f_1, \cdots, f_s >$. A *Gröbner basis* of the ideal

$< f_1, \cdots, f_s >$ describes the variety $V_{\mathcal{F}}$. For a detailed description of Gröbner bases and varieties, we refer the reader to [BeWe] and [COX].

Let $\mathcal{A} = \{a_1(X), \cdots, a_u(X)\} \in \mathbb{F}_q[X]^u$ and $\mathcal{B} = \{b_1(X), \cdots, b_u(X)\} \in \mathbb{F}_q[X]^u$ be two sets of polynomials. We shall say that these two sets are *linear-equivalent*, denoted $\mathcal{A} \equiv_L \mathcal{B}$, if there exists $S \in GL_n(\mathbb{F}_q)$ such that $b_i(X) = a_i(XS)$ for all $i, 1 \leq i \leq u$. We call such a matrix a *linear equivalence matrix between $\mathcal{A}$ and $\mathcal{B}$*. In the sequel, for convenience, we shall denote these equations by $\mathcal{B}(X) = \mathcal{A}(XS)$. The *Polynomial Linear Equivalence* (PLE) problem is then the problem of finding a linear equivalence matrix between $\mathcal{A}$ and $\mathcal{B}$, if any.

A natural extension is to consider bijective affine mappings over the $\mathbb{F}_q$-vector space $\mathbb{F}_q^n$. We shall say that two sets of polynomials $\mathcal{A}$ and $\mathcal{B}$ are *affine-equivalent*, denoted $\mathcal{A} \equiv_A \mathcal{B}$, if there exists $(S, T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$ such that $\mathcal{A}(X) = \mathcal{B}(XS + T)$. We call such a pair *an affine equivalence pair between $\mathcal{A}$ and $\mathcal{B}$*, S being the *linear part* of this pair and T being its *affine part*. The *Polynomial Affine Equivalence* (PAE) problem is then the problem of finding an affine equivalence pair between $\mathcal{A}$ and $\mathcal{B}$, if any.

This last problem was first introduced in [P] under the name *Isomorphism of Polynomials with one secret* problem, in reference to the well known graph isomorphism problem. We believe that this name is not well suited. Remember that two graphs are said to be isomorphic if and only if they are identical after a permutation of the vertices of one of the graphs. In such a setting, isomorphism is defined by a permutation and permutations are a special kind of bijective mappings. The problems which are addressed in [P] and here are much more general than the one of finding a permutation between two sets of polynomials. For this reason, we think that the name PLE and PAE we chose are better suited. Moreover, PLE and PAE are equivalence relations, as can be seen easily.

As pointed out by Geiselmann *et al.* in [GMS], it makes a difference whether the relations $\equiv_L$ and $\equiv_A$ are checked over $\mathbb{F}_q[X]$ or over $\mathbb{F}_q[X]/ < x_i^q - x_i >_{1 \leq i \leq n}$. Indeed, if $\mathcal{A} \equiv_A \mathcal{B}$(or $\mathcal{A} \equiv_L \mathcal{B}$) over $\mathbb{F}_q[X]$ then $\mathcal{A} \equiv_A \mathcal{B}$(or $\mathcal{A} \equiv_L \mathcal{B}$) over $\mathbb{F}_q[X]/ < x_i^q - x_i >_{1 \leq i \leq n}$ but the converse is not always true. In this paper, we only work with polynomials over $\mathbb{F}_q[X]/ < x_i^q - x_i >_{1 \leq i \leq n}$, since it appears to us to be the most natural space where cryptographic applications can be designed.

3 General properties of polynomial equivalence

We quote here properties of the polynomial equivalence. Proofs can be found in appendix A.

3.1 Structural Properties

For a polynomial $p \in \mathbb{F}_q[X]$, we shall denote by $p^{(d)}$ the terms of total degree d of this polynomial and by $p^{(\tilde{d})}$ his terms of highest total degree $\tilde{d}$. By extension, we shall denote

by $\mathcal{A}^{(d)} = \{a_1^{(d)}(X), \cdots, a_u^{(d)}(X)\}$ and by $\mathcal{B}^{(d)} = \{b_1^{(d)}(X), \cdots, b_u^{(d)}(X)\}$, the terms of total degree d of $\mathcal{A}$ and $\mathcal{B}$.

Note that if there exists an index i, $1 \leq i \leq u$, for which a_i and b_i do not have the same degree, then $\mathcal{A} \not\equiv_L \mathcal{B}$.

Proposition 1. *Let $S \in GL_n(\mathbb{F}_q)$. Then $\mathcal{B}(X) = \mathcal{A}(XS) \Longleftrightarrow \mathcal{B}^{(j)}(X) = \mathcal{A}^{(j)}(XS)$ for all $j, 0 \leq j \leq \widetilde{D}$, where $\widetilde{D}$ is the maximum total degree of the polynomials of $\mathcal{A}$ and $\mathcal{B}$.*

Proposition 2. *Let $\alpha \in \mathbb{F}_q$, $\mathcal{A}_\alpha = \{a_1(\alpha X), \cdots, a_u(\alpha X)\}$, $\mathcal{B}_\alpha = \{b_1(\alpha X), \cdots, b_u(\alpha X)\}$ be sets of polynomials and $S \in GL_n(\mathbb{F}_q)$. Then $\mathcal{B}(X) = \mathcal{A}(XS) \Longleftrightarrow \mathcal{B}_\alpha(X) = \mathcal{A}_\alpha(XS)$, for all α in $\mathbb{F}_q$.*

3.2 Geometrical Properties

In the sequel, we denote by $V_\mathcal{A}$ and $V_\mathcal{B}$ the varieties associated to $\mathcal{A}$ and $\mathcal{B}$.

Property 1. *Let $(S,T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$. If $\mathcal{B}(X) = \mathcal{A}(XS+T)$ then $V_\mathcal{A} = V_\mathcal{B}S + T$, with $V_\mathcal{B}S + T = \{v_\mathcal{B}S + T : v_\mathcal{B} \in V_\mathcal{B}\}$.*

Corollary 3.1. *Let $S \in GL_n(\mathbb{F}_q)$. If $\mathcal{B}(X) = \mathcal{A}(XS)$ then $V_\mathcal{A} = V_\mathcal{B}S$.*

By property 1 and corollary 3.1, we have that $\mathcal{A} \equiv_A \mathcal{B}$ or $\mathcal{A} \equiv_L \mathcal{B}$ implies $|V_\mathcal{A}| = |V_\mathcal{B}|$.

Property 2. *Let $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ be sets of polynomials, $V_\mathcal{A}, V_\mathcal{B}, V_\mathcal{C}, V_\mathcal{D}$ be the varieties associated to these sets and $(S,T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$.*

$$(\mathcal{B}(X) = \mathcal{A}(XS+T) \text{ and } \mathcal{D}(X) = \mathcal{C}(XS+T)) \Longrightarrow V_\mathcal{A} \cap V_\mathcal{C} = (V_\mathcal{B} \cap V_\mathcal{D})S + T.$$

By adding the field equations $\{x_1^q - x_1, \cdots, x_n^q - x_n\}$ to a set of polynomials $\mathcal{A}$, we change the geometry of the solutions. In particular, the variety associated to this new set is equal to $V_\mathcal{A} \cap \mathbb{F}_q^n$, a subset of $\mathbb{F}_q^n$ and not of $\bar{\mathbb{F}}_q^n$. Hence, we have a finite number of points in this variety.

Corollary 3.2. *Let $(S,T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$. If $\mathcal{B}(X) = \mathcal{A}(XS+T)$ then $V_\mathcal{A} \cap \mathbb{F}_q^n = (V_\mathcal{B} \cap \mathbb{F}_q^n)S + T$.*

By using structural properties of the affine equivalence relation, we get:

Proposition 3. *Let $(S,T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$.*
If $\mathcal{B}(X) = \mathcal{A}(XS+T)$ then $V_{\mathcal{A}_{1,p}} \cap \mathbb{F}_q^n = (V_{\mathcal{B}_{1,p}} \cap \mathbb{F}_q^n)S + T$ for any fixed $p \in \mathbb{F}_q^n$, $V_{\mathcal{A}_{1,p}}$ being the variety associated to $< a_1(X) - b_1(p), \cdots, a_u(X) - b_u(p) >$ and $V_{\mathcal{B}_{1,p}}$ the variety associated to $< b_1(X) - b_1(p), \cdots, b_u(X) - b_u(p) >$.

Remark 3.1. *The results of property 2, corollary 3.2 and proposition 3 are also true for the linear equivalence relation.*

For proposition 3, we have only used the properties of the affine equivalence relation. We get the next proposition by using particular properties of the linear equivalence relation.

Proposition 4. *Let* $(\alpha, p) \in \mathbb{F}_q \times \mathbb{F}_q^n, \mathcal{A}_{\alpha,p} = \{a_1(\alpha X) - b_1(\alpha p), \cdots, a_u(\alpha X) - b_u(\alpha p)\}$, $\mathcal{B}_{\alpha,p} = \{b_1(\alpha X) - b_1(\alpha p), \cdots, b_u(\alpha X) - b_u(\alpha p)\}, U \subseteq \mathbb{F}_q$ *and* $S \in GL_n(\mathbb{F}_q)$.

If $\mathcal{B}(X) = \mathcal{A}(XS)$ *then* $\bar{V}_{\mathcal{A}_{U,p}} = \bar{V}_{\mathcal{B}_{U,p}} S$, *for any fixed* $p \in \mathbb{F}_q^n$, *with:*

$\bar{V}_{\mathcal{A}_{U,p}} = (\cap_{\alpha \in U} V_{\mathcal{A}_{\alpha,p}}) \cap (\cap_{1 \leq d \leq \tilde{D}} V_{\mathcal{A}^{(d)}}) \cap \mathbb{F}_q^n$ *and* $\bar{V}_{\mathcal{B}_{U,p}} = (\cap_{\alpha \in U} V_{\mathcal{B}_{\alpha,p}}) \cap (\cap_{1 \leq d \leq \tilde{D}} V_{\mathcal{B}^{(d)}}) \cap \mathbb{F}_q^n$,

$\widetilde{D}$ *being the maximum total degree of the polynomials of* $\mathcal{A}$ *and* $\mathcal{B}$.

4 Polynomial linear equivalence algorithms

In this section, we present two new algorithms for solving the PLE problem. The first uses a link between this problem and that of finding the common zeroes of a set of polynomials. The properties given in 3.2 are used to design the second algorithm. We conclude this section by exhibiting instances which can be solved in deterministic polynomial time.

4.1 Our first algorithm

The basic idea follows [GMS]. We know that when two sets of polynomials $\mathcal{A} = \{a_1(X), \cdots, a_u(X)\}$ and $\mathcal{B} = \{b_1(X), \cdots, b_u(X)\}$ are linear-equivalent, then the evaluation of the b_is on some vector $p \in \mathbb{F}_q^n$ is equal to the evaluation of the a_is in $\widetilde{p}' = pS$, for some linear equivalence matrix S between $\mathcal{A}$ and $\mathcal{B}$. Knowledge of the pair $(p, \widetilde{p}')$ allows us to obtain n linear equations in the components of S. The main idea of this algorithm is to convert the search of these pairs into the solving of a non linear system of equations. For a vector $p \in \mathbb{F}_q^n$, we notice that the variety associated to the ideal $< \{a_i(X) - b_i(p)\}_{1 \leq i \leq u}, \{x_i^q - x_i\}_{1 \leq i \leq n} >$ gives all the vectors $p' \in \mathbb{F}_q^n$ such that $\mathcal{B}(p) = \mathcal{A}(p')$. We know that there exists a unique vector $\widetilde{p}'$ in this variety such that $\widetilde{p}' = pS$. When the polynomial mapping $\mathcal{A}$ is bijective, the variety above gives exactly one such vector. But it is not the case in general and in order to improve the effectiveness of the algorithm, we must construct varieties whose cardinalities are as small as possible. In order to do that, we will use the properties of 3.1. More precisely, according to proposition 1 and 2, if there exists a linear equivalence matrix S between $\mathcal{A}$ and $\mathcal{B}$, then we have the following:

$$\exists (p, \widetilde{p}') \in \mathbb{F}_q^n \times \mathbb{F}_q^n, \mathcal{B}(p) = \mathcal{A}(\widetilde{p}') \Longrightarrow \begin{cases} \mathcal{B}^{(j)}(p) = \mathcal{A}^{(j)}(\widetilde{p}'), \forall 0 \leq j \leq \widetilde{D}, \\ \mathcal{B}_\alpha(p) = \mathcal{A}_\alpha(\widetilde{p}'), \forall \alpha \in U \subseteq \mathbb{F}_q, \end{cases} \quad (I)$$

where $\widetilde{D}$ is the maximum total degree of the polynomials of $\mathcal{A}$ and $\mathcal{B}$.
We now give an algebraic interpretation of these constraints. Let α be in $\mathbb{F}_q$. We shall denote by $I_{\alpha,p} =< \{a_i(\alpha X) - b_i(\alpha p)\}_{1 \leq i \leq u}, \{x_i^q - x_i\}_{1 \leq i \leq n} >$ and by $V_{\alpha,p}$ the variety associated to this ideal. We also set $I_p^{(j)} =< \{a_i^{(j)}(X) - b_i^{(j)}(p)\}_{1 \leq i \leq u}, \{x_i^q - x_i\}_{1 \leq i \leq n} >$ and call $V_p^{(j)}$ the variety associated to this ideal. Finally $\bar{V}_{U,p} = (\cap_{\alpha \in U \cup \{1\}} V_{\alpha,p}) \cap (\cap_{1 \leq j \leq \widetilde{D}} V_p^{(j)})$ denotes the vectors $p' \in \mathbb{F}_q^n$ such that $\mathcal{A}(p') = \mathcal{B}(p)$ and achieve the constraints (I). With these

notations, we point out that if there exists a vector $p \in \mathbb{F}_q^n$ such that $\bar{V}_{U,p} = \emptyset$ for some $U \subseteq \mathbb{F}_q$ then, $\mathcal{A} \not\equiv_L \mathcal{B}$.

Idea of the algorithm
From the polynomials given in input, we construct sets $L_j, 1 \leq j \leq n$ such that each L_j contains the j-th row of candidates for the linear equivalence matrices between the two inputs. When such a matrix doesn't exist, the algorithm returns $\emptyset$. Let $\{e_j\}_{1 \leq j \leq n}$ be the n vectors of the canonical basis of $\mathbb{F}_q^n$. The algorithm is the following:

Algorithm A
Input: Two sets of polynomials $\mathcal{A}$ and $\mathcal{B}$.
Output: A linear equivalence matrix between $\mathcal{A}$ and $\mathcal{B}$, if any and $\emptyset$ otherwise.
For j from 1 to n **do**
Choose $U \subseteq \mathbb{F}_q$ randomly
Compute $\bar{V}_{U,e_j}$

If $\bar{V}_{U,e_j} \neq \emptyset$ **then** $L_j \leftarrow \bar{V}_{U,e_j}$ **Else Return** $\emptyset$

EndFor
$S \leftarrow SeekRows(\mathcal{A}, \mathcal{B}, \{L_1, \cdots, L_n\})$
Return S

For all $j, 1 \leq j \leq n$ the elements of L_j are candidates for the j-th row of a linear equivalence matrix between $\mathcal{A}$ and $\mathcal{B}$.

Remark 4.1. *If $\mathcal{A} \equiv_L \mathcal{B}$ and if $\mathcal{A}$ or $\mathcal{B}$ is a bijection*[3] *then there exists a unique linear equivalence matrix between these two sets. Indeed, for all j, $1 \leq j \leq n$, L_j only contains the j-th row of this matrix.*

The function *SeekRows* outputs a linear equivalence matrix between $\mathcal{A}$ and $\mathcal{B}$, if any, and $\emptyset$ otherwise. To recover this matrix, if such a matrix exists, it checks for all the invertible matrices than can be constructed from the sets $L_1, \cdots, L_n$. We propose in appendix B an improvement of this function.

Remark 4.2. *The bigger the size of the subset $U \subseteq \mathbb{F}_q$, the better algorithm A is. Indeed, the more equations you have, the faster are done the calculations of the Groëbner bases [Fa] and the smaller the number of candidates in the varieties computed are. But remember that the generation of the equations must be efficiently done.*

Complexity
The complexity of calculation of a Gröbner basis depends on the maximum degree of the polynomials occurring during this computation [BeWe]. This parameter depends on the set of polynomials but for polynomials which have a finite number of zeroes (varieties are so-called 0-dimensional varieties), which always occurs in cryptographic applications, it can be bounded from above by $\mathcal{O}(D^n)$ (see [BeWe] *p.513*). At each step $j, 1 \leq j \leq n$ the varieties

[3] When $\mathcal{A} \equiv_L \mathcal{B}$ and if one of the inputs is bijective then the other is also bijective

$\bar{V}_{U,e_j}$ computed have a finite number of points. Hence, when one of the inputs is a bijection, the complexity of this algorithm is $\mathcal{O}(nD^n)$.
In the general case, the function *SeekRows* checks the invertibility (by Gaussian elimination) of at most $\prod_{i=1}^{n} |L_i|$ matrices. Finally, for generic instances, this algorithm has a complexity of $\mathcal{O}(nD^n + n^3 \prod_{i=1}^{n} |L_i|)$. Our algorithm recovers in fact all the linear equivalence matrices between the two sets of polynomials. Indeed, this complexity is exactly the one of finding them all.

Previous work

To our knowledge, the only work done on this subject is presented by Geiselmann *et al.* in [GMS]. For a detailed description of their algorithm, we refer the reader to this article. We briefly recall in this part the principle of the algorithm proposed by these authors. We point out that it was dedicated for the PAE problem, but it can be easily adapted to the PLE problem. In this setting, the main idea is to remark that if $l \in \mathbb{F}_q^n$ is the j-th row of a linear equivalence matrix between two sets of polynomials $\mathcal{A}$ and $\mathcal{B}$, then $\mathcal{B}_\alpha(e_j) = \mathcal{A}_\alpha(l)$ for all $\alpha \in U \subseteq \mathbb{F}_q$. An exhaustive search among the vectors $l \in \mathbb{F}_q^n$ is then performed to recover these candidates.
The set $\{l \in \mathbb{F}_q^n : \mathcal{B}_\alpha(e_j) = \mathcal{A}_\alpha(l), \forall \alpha \in U\}$ is equal to $\cap_{\alpha \in U} V_{\alpha,e_j}$. Hence, we have substituted in our algorithm the exhaustive search of the elements of $\{l \in \mathbb{F}_q^n : \mathcal{B}_\alpha(e_j) = \mathcal{A}_\alpha(l), \forall \alpha \in U\}$ by the computation of a variety. In the worst case, the theoretical complexity of computing V_{α,e_j} is $\mathcal{O}(D^n)$, where D is the maximum degree of the polynomials of $\mathcal{A}$. This must be compared with the complexity $\mathcal{O}(q^n)$ of the exhaustive search of [GMS]. But the complexity of computing Gröbner bases depends in practice very much on the algorithm used and an efficient software, such as *fgb*, behaves much better than in the worst case[Fa]. In addition, by investigating structural properties of the PLE problem, we have also added new constrains which permit to decrease the size of the set of candidates and so to increase the efficiency of our algorithm.

4.2 A second algorithm

This algorithm is more particularly dedicated to sets of polynomials which are not bijective. The main idea is to use geometrical properties of the linear equivalence relation. According to corollary 3.1, when $\mathcal{A} \equiv_L \mathcal{B}$ the varieties $V_\mathcal{A}$ and $V_\mathcal{B}$ are such that $V_\mathcal{A} = V_\mathcal{B} S$, for some linear equivalence matrix S between $\mathcal{A}$ and $\mathcal{B}$. Hence, for each $v_\mathcal{B} \in V_\mathcal{B}$ there exists a unique vector $\widetilde{v}_\mathcal{A} \in V_\mathcal{A}$ such that $\widetilde{v}_\mathcal{A} = v_\mathcal{B} S$. According to proposition 4 the pair $(\widetilde{v}_\mathcal{A}, v_\mathcal{B})$ lies in $\bar{V}_{\mathcal{A}_U} \times \bar{V}_{\mathcal{B}_U}$, with:

$$\bar{V}_{\mathcal{A}_U} = (\cap_{\alpha \in U \cup \{1\}} V_{\mathcal{A}_\alpha}) \cap (\cap_{1 \leq d \leq \widetilde{D}} V_{\mathcal{A}^{(d)}}) \cap \mathbb{F}_q^n,$$
$$\bar{V}_{\mathcal{B}_U} = (\cap_{\alpha \in U \cup \{1\}} V_{\mathcal{B}_\alpha}) \cap (\cap_{1 \leq d \leq \widetilde{D}} V_{\mathcal{B}^{(d)}}) \cap \mathbb{F}_q^n,$$

where $\widetilde{D}$ is the maximum degree of the polynomials of $\mathcal{A}$ and $\mathcal{B}$ and U is a subset of $\mathbb{F}_q$.
This property permits to improve the search of the pair $(\widetilde{v}_\mathcal{A}, v_\mathcal{B})$ since it can be done in $\bar{V}_{\mathcal{A}_U} \times \bar{V}_{\mathcal{B}_U}$, a subset of $V_\mathcal{A} \times V_\mathcal{B}$. Knowledge of the pairs $\{(\widetilde{v}_\mathcal{A}, v_\mathcal{B}), v_\mathcal{B} \in \bar{V}_{\mathcal{B}_U}\}$ allows us to get $n * dim_{\mathbb{F}_q}(Span(\bar{V}_{\mathcal{B}_U}))$ linearly independent equations in the components of S.

It could be that the number of equations is not sufficient to recover S. Let p be a vector which is not in $Span(V_{\mathcal{B}})$. If $\mathcal{A} \equiv_L \mathcal{B}$, we have, according to proposition 3, $V_{\mathcal{A}} = V_{\mathcal{B}}S$ but also $V_{\mathcal{A}_{1,p}} = V_{\mathcal{B}_{1,p}}S$. For each vector $v \in V_{\mathcal{B}_{1,p}}$, we also have a unique vector $\widetilde{v} \in V_{\mathcal{A}_{1,p}}$ such that $\widetilde{v} = vS$. As explained in proposition 4, the search of the suitable pairs can be done on a subset of $V_{\mathcal{A}_{1,p}} \times V_{\mathcal{B}_{1,p}}$ and more precisely in $\bar{V}_{\mathcal{A}_{U,p}} \times \bar{V}_{\mathcal{B}_{U,p}}$, with:

$$\bar{V}_{\mathcal{A}_{U,p}} = (\cap_{\alpha \in U \cup \{1\}} V_{\mathcal{A}_{\alpha,p}}) \cap (\cap_{1 \leq d \leq \widetilde{D}} V_{\mathcal{A}^{(d)}}) \cap \mathbb{F}_q^n,$$
$$\bar{V}_{\mathcal{B}_{U,p}} = (\cap_{\alpha \in U \cup \{1\}} V_{\mathcal{B}_{\alpha,p}}) \cap (\cap_{1 \leq d \leq \widetilde{D}} V_{\mathcal{B}^{(d)}}) \cap \mathbb{F}_q^n,$$

where U is a subset of $\mathbb{F}_q$.
Hence, the pairs $\{(\widetilde{v}, v), v \in \bar{V}_{\mathcal{B}_{U,p}}\}$ allow us to get $n * dim_{\mathbb{F}_q}(Span(\bar{V}_{\mathcal{B}_{U,p}}))$ new linearly independent equations in the components of S, in addition to the equations always given by the pairs $\{(\widetilde{v}_{\mathcal{A}}, v_{\mathcal{B}}), v_{\mathcal{B}} \in \bar{V}_{\mathcal{B}_U}\}$. Since p is chosen not to lie in $Span(V_{\mathcal{B}})$, at least n of these news equations are linearly independent from the equations given by $\{(\widetilde{v}_{\mathcal{A}}, v_{\mathcal{B}}), v_{\mathcal{B}} \in \bar{V}_{\mathcal{B}_U}\}$. Note that, with these notations, if $\bar{V}_{\mathcal{A}_{U,p}} = \emptyset$ for some subset $U \subseteq \mathbb{F}_q$ and for some vector $p \in \mathbb{F}_q^n$ then $\mathcal{A} \not\equiv_L \mathcal{B}$.

Idea of the algorithm
As long as the number of equations in the components of the matrix we try to determine is not sufficient, we will compute, from the polynomials given in input, varieties $V^{1,k}$ and $V^{2,k}$, where $V^{1,k}$ is the $k-th$ variety $\bar{V}_{\mathcal{A}_{U,p}}$, for different choices of U and p. Variety $V^{2,k}$ is defined analogously (with respect to the set $\mathcal{B}$). Those varieties verify $V^{1,k} = V^{2,k}S$, for some linear equivalence matrix S between the two inputs, if any. When such a matrix doesn't exist, the algorithm returns $\emptyset$. The algorithm is the following:

Algorithm B
Input: Two sets of polynomials $\mathcal{A}$ and $\mathcal{B}$.
Output: A linear equivalence matrix between $\mathcal{A}$ and $\mathcal{B}$, if any and $\emptyset$ otherwise.
Initialization: $V^{1,1} = V^{2,1} = \cdots = V^{1,n} = V^{2,n} = \emptyset$,$P = \emptyset$, $cpt = 0$, $l = 1$.
While $cpt < n$ **do**

Choose $p \in \mathbb{F}_q^n \setminus P$ and $U \subseteq \mathbb{F}_q$ randomly

Compute $\bar{V}_{\mathcal{A}_{U,p}}$

If $\bar{V}_{\mathcal{A}_{U,p}} \neq \emptyset$ **then**

Compute $\bar{V}_{\mathcal{B}_{U,p}}$
$(V^{1,l}, V^{2,l}) \leftarrow (\bar{V}_{\mathcal{A}_{U,p}}, \bar{V}_{\mathcal{B}_{U,p}})$
$cpt \leftarrow cpt + dim_{\mathbb{F}_q}(Span(V^{2,l}))$
$l \leftarrow l + 1$
$P \leftarrow Span(P \cup V^{2,l})$

Else Return $\emptyset$

EndWhile
$S \leftarrow SeekMatrix(\{V^{1,k}, V^{2,k}\}_{1 \leq k \leq l})$
Return S

Remark 4.3. *Let* $d_k = dim_{\mathbb{F}_q}(Span(V^{2,k}))$. *At the end of the algorithm, we have* $\sum_{k=1}^{l} d_k \geq n$.

The function $SeekMatrix$ outputs a linear equivalence matrix between the two inputs, if any, and $\emptyset$ otherwise. It computes for each $k, 1 \leq k \leq$ a basis $B_{V^{2,k}} = \{v_1^{2,k}, \cdots, v_{d_k}^{2,k}\}$ of $Span(V^{2,k})$. For all the elements:

$$\{\big((v_1^{1,1}, v_1^{2,1}), \cdots, (v_{d_1}^{1,1}, v_{d_1}^{2,1})\big), \cdots, \big((v_1^{1,l}, v_1^{2,l}), \cdots, (v_{d_l}^{1,l}, v_{d_l}^{2,l})\big)\} \in (V^{1,1} \times B_{V^{2,1}})^{d_1} \times \cdots \times (V^{1,l} \times B_{V^{2,l}})^{d_l},$$

it checks if the linear system in the unknowns the components of a matrix M:

$$v_1^{2,1} M = v_1^{1,1}, \cdots, v_{d_1}^{2,1} M = v_{d_1}^{1,1}, \cdots, v_1^{2,l} M = v_1^{1,l}, \cdots, v_{d_l}^{2,l} M = v_{d_l}^{1,l} \quad (1)$$

is invertible and, if so, recovers this matrix. Finally, it checks if M is a linear equivalence matrix between $\mathcal{A}$ and $\mathcal{B}$.

Complexity
Let D be the maximum degree of the polynomials of $\mathcal{A}$ and $\mathcal{B}$. At each step $k, 1 \leq k \leq l$ of the while loop the varieties $V^{1,k}$ and $V^{2,k}$ have a finite number of points. Hence, the complexity of constructing these $2l$ sets is $\mathcal{O}(2lD^n)$. Let $N_k = |V^{1,k}|^{d_k}$. The function $SeekMatrix$ computes l bases of vector spaces, checks invertibility and solves at most $\prod_{k=1}^{l} N_k$ linear

systems. Moreover, the maximum number of steps performed by algorithm B is given by proposition 5:

Proposition 5. *If $\mathcal{A} \equiv_L \mathcal{B}$, then algorithm B performs at most n steps.*

Proof
Let $S \in GL_n(\mathbb{F}_q)$, such that $\mathcal{B}(X) = \mathcal{A}(XS)$. At the k-th step of algorithm B, the vector p is chosen in $\mathbb{F}_q^n \setminus Span(\cup_{0<r<k} V^{2,r})$. Therefore, the vector pS is linearly independent from the vectors $\{vS : v \in \cup_{0<r<k} V^{2,r}\}$. Hence, at each step of the while loop, algorithm B gives at least n new equations which are linearly independent from the equations given by $\{(vS, v) : v \in \cup_{0<r<k} V^{2,r}\}$. □

Finally, the complexity of this algorithm is $\mathcal{O}(2nD^n + n^6 \prod_{k=1}^{n} N_k)$. Note that this algorithm recovers in fact all the linear equivalence matrices between two sets of polynomials. This complexity is again exactly the one of finding them all.

Remark 4.4. *N_k represents an upper bound on the complexity of finding all subsets of d_k independent vectors of $V^{1,k}$. The value of N_k given above comes from enumerating* all *subsets of d_k vectors in $V^{1,k}$. As we are interested only in subsets of* independent *vectors, to construct such a set of vectors, one chooses the first vector - say v - at random, then the next one in $V^{1,k} \setminus (Span(\{v\}) \cap V^{1,k})$, and so on. Thus, in order to evaluate the complexity of finding all bases of cardinality d_k in $V^{1,k}$, it would be more accurate to set $N_k = \prod_{r=1}^{d_k} (|V^{1,k}| - q^{r-1})$. This might change the estimate on the complexity of function $SeekMatrix$, especially when the cardinality of $V^{1,k}$ is small.*

The number of steps given in proposition 5 is the one occuring when $d_k = 1$, $\forall 1 \leq k \leq l$, in which case $l = n$. It is clear that in practice this number will almost always be much less. Indeed, the important point of proposition 5 is that the algorithm actually terminates.

Comparison with algorithm A
For each $j, 1 \leq j \leq n$ the variety $\bar{V}_{U,e_j}$ computed in algorithm A is equal to the variety $\bar{V}_{\mathcal{A}_{U,e_j}}$ in algorithm B. By construction, these two varieties contain a vector $l_j \in \mathbb{F}_q^n$ which is the j-th row of some linear equivalence matrix S between $\mathcal{A}$ and $\mathcal{B}$. In A, we focus only on recovering the rows of a linear equivalence matrix. Hence, we have only to compute for each j the variety $\bar{V}_{U,e_j}$, which contains the candidates for the j-th row of this matrix. In B, when we compute $\bar{V}_{\mathcal{A}_{U,e_j}}$ and $\bar{V}_{\mathcal{B}_{U,e_j}}$, we also find the pair (l_j, e_j) but in addition we try to recover other pairs $(\widetilde{v}, v) \in \bar{V}_{\mathcal{A}_{U,e_j}} \times \bar{V}_{\mathcal{B}_{U,e_j}}$ such that $\widetilde{v} = vS$.

Selection Strategy
In fact, these two algorithms are complementary and in order to minimize the number of varieties computed, you can use the following strategy.
Start with algorithm A and for each j,$1 \leq j \leq n$ of the *for* loop, compute $t = \sum_{k=1}^{j} dim_{\mathbb{F}_q}(Span(V_{U,e_k}))$.
If $t \geq n$ and $2j < n$ then stop the execution of A, compute for all k, $1 \leq k \leq j$ the varieties $V_{\mathcal{B}_{U,e_k}}$ and recover a linear equivalence matrix with the function

$SeekMatrix(\{V_{U,e_k}, V_{\mathcal{B}_{U,e_k}}\}_{1\leq k\leq j})$.
Else continue the execution of algorithm A.
Notice that $2j$ is the number of varieties computed in algorithm B, in the case when the set of vectors chosen during the *while* loop of algorithm B were $\{e_1, \cdots, e_j\}$.

4.3 Weak Instances of PLE

From both a practical and theoretical point of view, it is revelant to be able to identify the instances which can be solved by a deterministic polynomial time algorithm. It is of major interest when this problem is used in cryptography. In this part, we present instances of the PLE problem admitting a deterministic polynomial time algorithm.

When restricting the inputs of the PLE problem to sets of polynomials of degree one, the problem can be reformulated as follows:
Input: Two matrices A and B in $\mathcal{M}_{u,n}(\mathbb{F}_q)$.
Question: Find if there exists a matrix $S \in GL_n(\mathbb{F}_q)$ such that $B = SA$.

If one of the inputs matrices is invertible[4], BA^{-1} is the unique solution to this problem

More generally, consider two sets of polynomials $\mathcal{A}$ and $\mathcal{B}$. According to proposition 1, we know that if there exists $S \in GL_n(\mathbb{F}_q)$ such that $\mathcal{B}(X) = \mathcal{A}(XS)$ then $\mathcal{B}^{(1)}(X) = \mathcal{A}^{(1)}(XS)$. In the other direction, if we know that $\mathcal{A}^{(1)} \equiv_L \mathcal{B}^{(1)}$ and the mapping $\mathcal{A}^{(1)}$ or $\mathcal{B}^{(1)}$ is bijective then the unique linear equivalence matrix between $\mathcal{A}^{(1)}$ and $\mathcal{B}^{(1)}$ is $S^{(1)} = B^{(1)}(A^{(1)})^{-1}$, where $A^{(1)}$ and $B^{(1)}$ are the matrices representing the linear mapping $\mathcal{A}^{(1)}$ and $\mathcal{B}^{(1)}$. Since S is also a linear equivalence matrix between $\mathcal{A}^{(1)}$ and $\mathcal{B}^{(1)}$, we have $S^{(1)} = S$. The unique linear equivalence matrix between $\mathcal{A}^{(1)}$ and $\mathcal{B}^{(1)}$ is a linear equivalence matrix between $\mathcal{A}$ and $\mathcal{B}$. Consequently, when the linear part of one of the inputs of the PLE problem is bijective then we can find a solution by performing very basic linear algebra operations.

5 Polynomial affine equivalence algorithms

5.1 General Method

Since the PAE problem is very similar to the PLE problem, it seems natural to try to reuse the algorithms described in section 4. A straightforward way to do this is:

For a in $\mathbb{F}_q^n$,
Try to find by algorithm A or B an $S \in GL_n(\mathbb{F}_q)$ such that $\mathcal{B}(X) = \mathcal{A}(XS + a)$.
If so, return (S, a).

This approach - which we shall call *general method* in the sequel - adds a factor q^n to the complexity of the algorithms A or B of section 4. Another method can be derived from the linear case by using an idea of 4.1, as follows.

[4]Remark that if $\mathcal{A} \equiv_L \mathcal{B}$, then if A(*resp.* B) is invertible then B(*resp.* A) is also invertible !

5.2 Generalization of algorithm A

We present here the affine version of algorithm A presented in 4.1. When $\mathcal{A} \equiv_A \mathcal{B}$, then the evaluation of the b_is on some vector $p \in \mathbb{F}_q^n$ is equal to the evaluation of the a_is in $\widetilde{p}' = pS + T$, for some affine equivalence pair (S, T) between $\mathcal{A}$ and $\mathcal{B}$. Hence, knowledge of the pair $(p, \widetilde{p}')$ allows us to obtain n linear equations in the components of S and T. In order to convert the search of this pair into the resolution of a non linear system of equations, we set $I_p =< \{a_i(X) - b_i(p)\}_{1 \leq i \leq u}, \{x_i^q - x_i\}_{1 \leq i \leq n} >$ and denote V_p the variety associated to this ideal. Let e_0 be the null vector of $\mathbb{F}_q^n$ and $\{e_j\}_{1 \leq j \leq n}$ be the n vectors of the canonical basis of $\mathbb{F}_q^n$. The algorithm is the following:

Algorithm A'
Input: Two sets of polynomials $\mathcal{A}$ and $\mathcal{B}$.
Output: An affine equivalence pair between $\mathcal{A}$ and $\mathcal{B}$, if any and $\emptyset$ otherwise.
Compute V_{e_0}

If $V_{e_0} \neq \emptyset$ **then**

$L_0 \leftarrow V_{e_0}$

For j from 1 to n **do**

Compute V_{e_j}

If $V_{e_j} \neq \emptyset$ **then** $L_j \leftarrow \{v_{e_j} - l_0 : (v_{e_j}, l_0) \in V_{e_j} \times L_0\}$ **Else Return** $\emptyset$

EndFor

Else Return $\emptyset$

$(S, T) \leftarrow Seek(\mathcal{A}, \mathcal{B}, \{L_0, \cdots, L_n\})$
Return (S, T)

L_0 is the set of candidates for the affine part of an affine equivalence pair between $\mathcal{A}$ and $\mathcal{B}$ and for all $j, 1 \leq j \leq n$ the elements of L_j are candidates for the j-th row of the linear part of an affine equivalence pair. To recover this pair, if such a pair exists, the function *Seek* checks for all the vectors in L_0 and for all the matrices than can be constructed from the sets $L_1, \cdots, L_n$. The improvement proposed in Appendix B can also be adapted to this function.

Remark 5.1. *If $\mathcal{A} \equiv_L \mathcal{B}$ and if $\mathcal{A}$ or $\mathcal{B}$ is a bijection then there exists a unique affine equivalence pair between these two sets.*

Complexity of algorithm A'
Let D be the maximum degree of the polynomials of $\mathcal{A}$. When one of the inputs is a bijection, the complexity A' is $\mathcal{O}((n+1)D^n)$. In the general case, the complexity is $\mathcal{O}((n+1)D^n + (n+1)^3 \prod_{i=0}^{n} |L_i|)$.

5.3 Generalization of algorithm B

The adaptation of algorithm B to the affine case is straightforward and is omitted.

5.4 Selection strategy

When u is "small" compared to n, it is very important to be able to decrease the size of the varieties computed in A'or in B'. Unfortunately, contrary to the linear case, the structural properties of the affine equivalence relation are useless in this context. In this case, we think that the general method together with algorithm B is the best choice to do. Indeed, the general method transforms a PAE problem into a PLE problem and even if it adds a factor q^n to the complexity of algorithm B, it allows to decrease the search space of the linear part of an equivalence pair. In the other case, one can use the following strategy to minimize the number of varieties computed.
Start with algorithm A' and for each j,$1 \leq j \leq n$ of the *for* loop, compute $t = \sum_{k=0}^{j} dim_{\mathbb{F}_q}(Span(V_{e_k}))$.
If $t \geq n+1$ and $2j < n+1$, stop the execution of A', compute for all k, $0 \leq k \leq j$ the varieties $V_{\mathcal{B}_{e_k}}$. Given $\{V_{e_k}\}_{0 \leq k \leq j}$ and $\{V_{\mathcal{B}_{e_k}}\}_{0 \leq k \leq j}$, you can recover an affine equivalence pair with a simple extension of the function $SeekMatrix$ described in section 4.2.
Else continue the execution of B'.

6 Applications

6.1 Security of cryptosystems based on PAE

Remember that we call PAE problem in this paper, the problem which is called isomorphism of polynomials with one secret in [P]. In this article, it has been shown how the PAE problem can be used to derive a signature and authentication scheme. We do not recall these constructions here, as our algorithms focus on the underlying problem which guarantees the security of these schemes. Let us recall the parameters for which the PAE problem was supposed intractable[P],[P1]. The two sets $\mathcal{A}$ and $\mathcal{B}$ are composed of $u \geq 2$ polynomials in n indeterminates of degree two whose coefficients lie in $\mathbb{F}_q$. The author recommends to choose the number of variables n and the size q of the field such that $q^{\sqrt{2}n^{3/2}} \geq 2^{64}$. We will now show that, in some cases, these parameters are far from being sufficient to achieve a reasonable level of security. In particular, in order to improve the efficiency of these schemes, the author suggests to restrict the affine equivalence to the linear equivalence. This restriction strongly acts on the safety of these schemes, since as explained in 4.3 the PLE problem has more properties than the PAE problem. Moreover, this problem admits a lot of instances which can be solved in deterministic polynomial time. Hence, without adding structural constraints on the shape of the polynomials of $\mathcal{A}$ and $\mathcal{B}$, which have not been given in the original design, these schemes are insecure. In order to avoid these weaknesses, the sets of polynomials $\mathcal{A}$ and $\mathcal{B}$ must be chosen in such a way that the linear parts $\mathcal{A}^{(1)}$ and $\mathcal{B}^{(1)}$ are not bijective. Even with these additional constraints, these parameters don't really give rise to difficult instances of the PLE problem and are not adapted to the design

of secure applications, as we shown in appendix D. Furthermore, the polynomial affine equivalence problem admits also instances for which the complexity of resolution is far from the cryptographically safe bounds. Experimental results are given in appendix D.

6.2 Chosen Ciphertext Attack

A large family of multivariate asymmetric encryption cryptosystems, like C^*[MI], HFE [P] and TTM [TTM] can be sketched as follows. Alice generates a set of polynomials $\mathcal{A} = \{a_1(x_1, \cdots, x_n), \cdots, a_u(x_1, \cdots, x_n)\} \subset \mathbb{F}_q[x_1, \cdots, x_n]^u$ in such a way that for all $c = (c_1, \cdots, c_u) \in \mathbb{F}_q^u$ there exists a unique solution [5] to the system $\{a_1(x_1, \cdots, x_n) - c_1 = 0, \cdots, a_u(x_1, \cdots, x_n) - c_u = 0\}$ and this solution can be efficiently computed. In order to hide $\mathcal{A}$, Alice chooses two pairs $(S,T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$ and $(U,V) \in GL_u(\mathbb{F}_q) \times \mathbb{F}_q^u$, computes $\mathcal{B}(X) = U \circ \mathcal{A}(XS+T)+V^t$, denoted by $\mathcal{B} = \{b_1(x_1, \cdots, x_n), \ldots, b_u(x_1, \cdots, x_n)\}$, and publishes $\mathcal{B}$. When Bob wants to encrypt a message $m = (m_1, \cdots, m_n) \in \mathbb{F}_q^n$, he computes $c = (b_1(m_1, \cdots, m_n), \cdots, b_u(m_1, \cdots, m_n))$ and sends it to Alice. After receiving $c = (c_1, \cdots, c_u)$, Alice computes the solution $c' = (c'_1, \cdots, c'_n)$ of the system $\mathcal{A}(X) - U^{-1}c^t + U^{-1}V^t = 0$ and recovers the message sent by computing $(c' - T)S^{-1} = m$.

An open question is to know whether or not these schemes remain secure if the set of polynomials $\mathcal{A}$ is public. In this situation, the security of these schemes relies not only on the difficulty of finding a common zero of a system of non linear equations but also on the difficulty of the Isomorphism of Polynomials (IP) problem (when $\mathcal{A}$ and $\mathcal{B}$ are given, the problem is to recover the pairs (S,T) and (U,V)). Until now, this question remains open since the best algorithm known to solve the IP problem[6] has a complexity of $O(q^{\frac{3n}{2}})$[CGP]. As pointed out in this paper, the PAE problem seems to be more difficult than the IP problem. Moreover, they have also shown that unless the polynomial hierarchy collapses, the PAE problem and the IP problem are not NP-hard. But contrary to the 'PAE problem, which is at least as difficult as the graph isomorphism problem, we would like to emphasize that there exists no theoretical evidence of the hardness of the IP problem. Hence, it is a natural question to ask whether the security of these schemes could be increased if Alice would choose $(U,V) = (I_u, \mathbf{0}_u)$[7] and publish $\mathcal{A}(X)$ and $\mathcal{B}(X) = \mathcal{A}(XS+T)$.

We now show that if the public key is generated in this way, an adversary is able to recover the secret pair $(S,T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$ with only $n+1$ queries to a deciphering oracle. Remark that due to the symmetry of the relation $\equiv_A$, we have $\mathcal{A}(X) = \mathcal{B}(XS'+T')$, with $S' = S^{-1}$ and $T' = -TS^{-1}$. Let $\{e_i\}_{1 \leq i \leq n}$, be the n canonical vectors of $\mathbb{F}_q^n$. In order to recover the secret vector T', an adversary sends $c_0 = (a_1(e_0), \cdots, a_u(e_0))$. The unique cleartext $m_0 \in \mathbb{F}_q^n$ corresponding to this ciphertext is such that $b_i(m_0) = a_i(e_0)$ for all i, $1 \leq i \leq u$. Hence, $m_0 = e_0S' + T' = T'$. To recover the j-th row of the matrix S',$1 \leq j \leq n$, an adversary sends $c_j = (a_1(e_j), \cdots, a_u(e_j))$. The cleartext $m_j \in \mathbb{F}_q^n$ corresponding to this ciphertext is such that $b_i(m_j) = a_i(e_j)$ for all i, $1 \leq i \leq u$. We then have $m_j = e_jS' + T'$

[5] or very few solutions

[6] This algorithm works only in the particular case when the sets of polynomials $\mathcal{A}$ and $\mathcal{B}$ are bijective.

[7] $\mathbf{0}_u$ is here the null vector of $\mathbb{F}_q^u$

and therefore the j-th row of the matrix S' is equal to $m_j - m_0$. Finally, knowledge of the pair (S', T') allows to recover easily the secret key (S, T) of Alice. Remark that when the cleartext is not unique, we obtain with this method not exactly the rows of the secret pair (S, T) but a list of candidates. Since for each row the number of candidates is not too big (otherwise Alice herself would not be able to decrypt), the secret affine pair can be recovered efficiently with the method described in [GMS] or with an extension of the method described in appendix B. Hence, the security of encryption schemes like HFE, can not be related to the difficulty of the PAE problem since in this situation the problem can be easily solved with the help of a deciphering oracle.

Let $(\mathcal{A}, U \circ \mathcal{A}(XS + T) + V^t)$ be the public-key of a multivariate encryption scheme. It is straightforward to see that if an adversary is able to recover the secret pair (U, V) then he can use the method described above to find the other pair (S, T). Hence, our method can be used in addition to an attack specifically designed to recover (U, V). Moreover, one sees at once that when the secret pair (S, T) is given then the pair (U, V) can be easily recovered. That is, when one of the two secret pairs is known, the other can be easily recovered. This is not the case for the underlying IP problem and so the problem considered in the security analysis of these schemes is weaker than the generic problem. It is left as an open problem whether or not the IP problem can be solved in deterministic polynomial time if we have access to a deciphering oracle.

7 Numerical results

Experimental results on the PLE problem

Conditions of the tests

We have generated a set of polynomials $\mathcal{A} = \{a_1(X), \cdots, a_u(X)\}$ with respect to the constraints given in 6.1 and we have chosen q and n such that $q^{\sqrt{2}n^{3/2}} \geq 2^{64}$. We have randomly chosen a matrix S in $GL_n(\mathbb{F}_q)$, we have computed $\mathcal{B} = \{a_1(XS), \cdots, a_u(XS)\}$ and we have tested one of the algorithms described in section 4 with these two sets of polynomials on a standard PC, using Magma software [Magma]. The results are quoted below:

n	u	$field$	$q^{\sqrt{2}n^{3/2}}$	$Algo$	$Time$
16	16	$\mathbb{F}_2$	2^{90}	A	≈ 4 *min.*
16	14	$\mathbb{F}_2$	2^{90}	A	≈ 7 *min.*
16	12	$\mathbb{F}_2$	2^{90}	B	≈ 18 *min.*
14	14	$\mathbb{F}_9$	2^{221}	A	≈ 7 *min.*
14	12	$\mathbb{F}_9$	2^{221}	A	≈ 15 *min.*

Experimental results on the PAE problem

Conditions of the tests

Let $\mathcal{A} = \{a_1(X), \cdots, a_u(X)\}$ be a set of polynomials chosen at random, q and n be chosen as above. We have randomly selected a pair $(S, T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$, we have computed

$\mathcal{B} = \{a_1(XS+T), \cdots, a_u(XS+T)\}$ and we have tested one of the algorithms described in section 5 and appendix C with these two sets of polynomials. We have quoted the results in the following table:

n	u	$field$	$q^{\sqrt{2n^{3/2}}}$	$Algo$	$Time$
16	16	$\mathbb{F}_2$	2^{66}	A'	≈ 5 *min.*
16	14	$\mathbb{F}_2$	2^{66}	A'	≈ 8 *min.*
16	12	$\mathbb{F}_2$	2^{66}	GM+B	≈ 20 *min.*
14	14	$\mathbb{F}_9$	2^{221}	A'	≈ 10 *min.*
14	12	$\mathbb{F}_9$	2^{221}	A'	≈ 20 *min.*

GM+B stands for General Method together with algorithm B.

Interpretation

Since we have chosen $u \approx n$, the cost of our algorithms is approximately the cost of computing several Gröbner Bases. Hence, we can exhibit a large number of instances illustrating the weakness of the security parameters given in [P]. We believe that the parameters chosen are significative of the behaviour of our algorithms.

When the two sets of polynomials lie in $\mathbb{F}_2[X]$, the efficiency of the algorithms dedicated to the PAE problem are similar to the ones dedicated to the PLE problem. Indeed, the PLE algorithms use the particular properties of the linear equivalence relation. But when the field is reduced to two elements, proposition 2 gives no information about the linear equivalence matrix. Whereas when the field is bigger, one can see that the PLE algorithms find a solution more quickly than the PAE algorithms.

Experimentally, it appears that algorithm B (*resp.* GM+B) is more efficient than algorithm A (*resp* A') when $n - u \geq 4$. In this situation algorithm B (*resp.* GM+B) computes only two varieties. For generic instances, we can consider that this relation among u and n can be used to select the better algorithm.

Some parameters propositions

For cryptographic applications, we strongly believe that u must be chosen approximately equal to n. In this setting, we know that there exists very few solutions to our problems. When u is small compared to n, there will probably be a large number of solution to these problems. Hence, an improved exhaustive search like local search method or test and trials method could work rather well.

For the PLE problem, we believe that the instances must be composed of homogeneous polynomials over $\mathbb{F}_2[X]$. In this setting, one can see at once that proposition 1 and proposition 2 give no information on the linear equivalence matrices. With such instances, we are in the same situation than for the PAE problem. For this problem, we have not found any particular weakness in the structure of the instances.

8 Conclusion

We have presented new approaches to the IP with one secret problem, which lead to the design of efficient algorithms. We studied the security of cryptosystems based on this problem and it appears that the usually suggested parameters often yield weak instances. Advises concerning parameters sizes are given at the end of the paper. We would like to point out that the complexities given throughout the paper suppose that the computations of the varieties involved are done by means of computing Gröbner bases, which is - to our knowledge - the best tool to date. In case a different algorithm is used, the complexities have to be modified accordingly.

References

[BeWe] T. Becker and V. Weispfenning. Gröbner Bases, A Computational Approach to Commutative Algebra. In cooperation with Heinz Kredel. Graduate Texts in Mathematics, 141. Springer-Verlag, New York, 1993.

[CGP] N. Courtois, L. Goubin, J. Patarin: Improved Algorithms for Isomorphism of Polynomials. Eurocrypt'98, Springer-Verlag, pp 184-200

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

[Fa] J.-C. Faugère: Algebraic cryptanalysis of HFE using Gröbner bases. INRIA report: RR-4738. Available from http://www.inria.fr/rrrt/rr-4738.html

[Fa99] J.-C. Faugère: A new efficient algorithm for computing Gröbner basis: F_4. Journal of pure and applied algebra, vol. 139, 1999, pp. 61–68.

[Fa02] J.-C. Faugère: A new efficient algorithm for computing Gröbner basis without reduction to zero: F_5. Proceedings of ISSAC, pages 75-83. ACM press, July 2002.

[FJ] J.-C. Faugère, Antoine Joux: Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. Crypto 2003, LNCS 2729, Springer-Verlag, August 2003.

[GMS] W. Geiselmann and W. Meier and R. Steinwandt: An Attack on the Isomorphisms of Polynomials Problem with One Secret Cryptology ePrint Archive: Report 2002/143. Available from http://eprint.iacr.org/2002/143

[Magma] http://magma.maths.usyd.edu.au/magma/

[MI] T. Matsumoto, H. Imai: Public Quadratic Polynomial-tuples for efficient signature-verification and message-encryption. Eurocrypt'88, Springer-Verlag, pp. 419-453.

[P] J. Patarin: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms. Eurocrypt'96, Springer-Verlag, pp. 33-48.

[P1] J. Patarin: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms - Extended version. http://www.cp8.com/sct/uk/partners/page/publi/eurocryptb.ps

[TTM] T.-T.Moh: A Fast Public Key System With Signature And Master Key Functions. CrypTEC'99, Hong Kong City University Press, pages 63-69, July 1999.

Appendix A

Proof of proposition 1

$\Rightarrow$

Suppose that $\mathcal{B}(X) = \mathcal{A}(XS)$, since the polynomials of $\mathcal{A}$ and $\mathcal{B}$ can be written uniquely as a sum of terms, we have $\sum_{j=0}^{\widetilde{D}} b_i^{(j)}(X) = \sum_{j=0}^{\widetilde{D}} a_i^{(j)}(XS)$ for all $i, 1 \leq i \leq u$. By equating the terms of total degree j, we get $\mathcal{B}^{(j)}(X) = \mathcal{A}^{(j)}(XS)$ for all j, $0 \leq j \leq \widetilde{D}$.

$\Leftarrow$

On the other direction, suppose that $\mathcal{B}^{(j)}(X) = \mathcal{A}^{(j)}(XS)$ for all j, $0 \leq j \leq \widetilde{D}$. We then have $b_i^{(j)}(X) = a_i^{(j)}(XS)$ for all $i, 1 \leq i \leq u$ and for all j, $0 \leq j \leq \widetilde{D}$. By summing the equalities, we get $\sum_{j=0}^{\widetilde{D}} b_i^{(j)}(X) = \sum_{j=0}^{\widetilde{D}} a_i^{(j)}(XS)$ for all $i, 1 \leq i \leq u$. Since the decomposition is unique, we have $\mathcal{B}(X) = \mathcal{A}(XS)$. □

The proof of proposition 2 is straightforward, and is omitted.

Proof of property 1

Suppose that there exists a pair $(S,T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$ such that $\mathcal{B}(X) = \mathcal{A}(XS+T)$.
We first prove $V_{\mathcal{B}}S + T \subseteq V_{\mathcal{A}}$. Let $v' = v_{\mathcal{B}}S + T$ be an element of $V_{\mathcal{B}}S + T$. According to the symmetry of the relation $\equiv_A$, we get $\mathcal{A}(v') = \mathcal{B}((v'-T)S^{-1}) = \mathcal{B}((v_{\mathcal{B}}S+T-T)S^{-1}) = \mathcal{B}(v_{\mathcal{B}})$. Since $v_{\mathcal{B}} \in V_{\mathcal{B}}$, one sees at once that $\mathcal{A}(v') = 0$. Hence v' lies in $V_{\mathcal{A}}$ and we have $V_{\mathcal{B}}S + T \subseteq V_{\mathcal{A}}$.
Now $V_{\mathcal{A}} \subseteq V_{\mathcal{B}}S + T$ is straightforward since, if v is an element of $V_{\mathcal{A}}$, we have $\mathcal{A}(v) = \mathcal{B}((v-T)S^{-1}) = 0$. Hence $(v-T)S^{-1}$ lies in $V_{\mathcal{B}}$, that is $v \in V_{\mathcal{B}}S + T$. □

Note that, since property 1 is true for all $T \in \mathbb{F}_q^n$ it is also true for $T = e_0$, the null vector of $\mathbb{F}_q^n$. Thus proving corollary 3.1.

Proof of property 2

Since $\mathcal{B}(X) = \mathcal{A}(XS+T)$ and $\mathcal{D}(X) = \mathcal{C}(XS+T)$ we get according to property 1, $V_{\mathcal{A}} = V_{\mathcal{B}}S + T$ and $V_{\mathcal{C}} = V_{\mathcal{D}}S + T$. Whence $V_{\mathcal{A}} \cap V_{\mathcal{C}} = (V_{\mathcal{B}}S+T) \cap (V_{\mathcal{D}}S+T) = (V_{\mathcal{B}} \cap V_{\mathcal{D}})S + T$.
□

Proof of corollary 3.2

Straightforward, since for all $(S,T) \in GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$, $Fro(X) = Fro(XS+T)$, with $Fro = \{x_1^q - x_1, \cdots, x_n^q - x_n\}$. Moreover, $V_{Fro} = \mathbb{F}_q^n$. □

Proof of proposition 3
On sees at once that if there exists a pair (S,T) of $GL_n(\mathbb{F}_q)\times\mathbb{F}_q^n$ such that $\mathcal{B}(X)=\mathcal{A}(XS+T)$ then for all fixed $p\in\mathbb{F}_q^n$, we have $\mathcal{B}_{1,p}(X)=\mathcal{A}_{1,p}(XS+T)$. Therefore, according to property 1 and corollary 3.2, we get $V_{\mathcal{A}_{1,p}}\cap\mathbb{F}_q^n=(V_{\mathcal{B}_{1,p}}\cap\mathbb{F}_q^n)S+T$ for any fixed $p\in\mathbb{F}_q^n$. □

Proof of proposition 4
As explained in the proof of proposition 3, if there exists a matrix S such that $\mathcal{B}(X)=\mathcal{A}(XS)$, then $\mathcal{B}_{1,p}(X)=\mathcal{A}_{1,p}(XS+T)$ for any fixed $p\in\mathbb{F}_q^n$. The linear equivalence relation have many more properties than the affine equivalence relation. According to propositions 1 and 2, it is straightforward to see that for a matrix $S\in GL_n(\mathbb{F}_q)$ and for any fixed $p\in\mathbb{F}_q^n$, we have:

$$\mathcal{B}(X)=\mathcal{A}(XS)\Rightarrow\left\{\begin{array}{l}\mathcal{B}_{\alpha,p}(X)=\mathcal{A}_{\alpha,p}(XS),\ \text{for all } \alpha \text{ in } U.\\ \mathcal{B}^{(j)}_{\alpha,p}(X)=\mathcal{A}^{(j)}_{\alpha,p}(XS),\ \text{for all } j, 0\le j\le\widetilde{D}.\end{array}\right.$$

Since for all j, $1\le j\le\widetilde{D}$, the polynomials of $\mathcal{A}^{(j)}_{\alpha,p}$ and $\mathcal{B}^{(j)}_{\alpha,p}$ are homogeneous, we have:

$$\begin{array}{rcl}\mathcal{A}^{(j)}_{\alpha,p} & = & \{(a_1(\alpha X)-a_1(\alpha p))^{(j)},\cdots,(a_u(\alpha X)-a_u(\alpha p))^{(j)}\}=\{a_1^{(j)}(\alpha X),\cdots,a_u^{(j)}(\alpha X)\}\\ & = & \{\alpha^j a_1^{(j)}(X),\cdots,\alpha^j a_u^{(j)}(X)\}=\alpha^j\mathcal{A}^{(j)}\\ \mathcal{B}^{(j)}_{\alpha,p} & = & \{(b_1(\alpha X)-b_1(\alpha p))^{(j)},\cdots,(b_u(\alpha X)-b_u(\alpha p))^{(j)}\}=\{b_1^{(j)}(\alpha X),\cdots,b_u^{(j)}(\alpha X)\}\\ & = & \{\alpha^j b_1^{(j)}(X),\cdots,\alpha^j b_u^{(j)}(X)\}=\alpha^j\mathcal{B}^{(j)}\end{array}$$

Therefore $\mathcal{B}^{(j)}(X)=\mathcal{A}^{(j)}(XS)\Leftrightarrow\mathcal{B}^{(j)}_{\alpha,p}(X)=\mathcal{A}^{(j)}_{\alpha,p}(XS)$ for all j, $1\le j\le\widetilde{D}$, for all $\alpha\in\mathbb{F}_q$ and for any fixed $p\in\mathbb{F}_q^n$. □

When $j=0$, remark that $\mathcal{A}^{(0)}_{\alpha,p}=\{(a_1(0)-b_1(\alpha p)),\cdots,(a_u(0)-b_u(\alpha p))\}$ and $\mathcal{B}^{(0)}_{\alpha,p}=\{(b_1(0)-b_1(\alpha p)),\cdots,(b_u(0)-b_u(\alpha p))\}$ which is not equal to $\mathcal{B}^{(0)}$ and $\mathcal{A}^{(0)}$.

Appendix B

Improvement of the function *SeekRows*
In [GMS], Geiselmann *et al.* propose a heuristic to improve the search of the good candidates. We propose here a slightly different heuristic which has the advantage to be easily parallelized.

Let $k\le n$ and $\{L_{i_1},\cdots L_{i_k}\}\subset\{L_1,\cdots,L_n\}$. If $(l_{i_1},\cdots,l_{i_k})\in L_{i_1}\times\cdots\times L_{i_k}$ are the l_{i_j}-th rows of some linear equivalence matrix S between $\mathcal{A}$ and $\mathcal{B}$, then $\mathcal{B}(\sum_{j=1}^k c_j l_{i_j})=\mathcal{A}(\sum_{j=1}^k c_j e_{i_j})$ for all $(c_1,\cdots,c_k)\in U\times\cdots\times U$, where $U\subseteq\mathbb{F}_q$. Therefore, from $L_{i_1},\cdots L_{i_k}$, we obtain new sets $\widetilde{L}_{i_1},\cdots,\widetilde{L}_{i_k}$ by selecting the k-tuples of $L_{i_1}\times\cdots\times L_{i_k}$ which achieve the conditions above. Similarly to $L_{i_1},\cdots L_{i_k}$ these new sets also contain for all j,$1\le j\le k$ the l_{i_j}-th rows of the linear equivalence matrices between $\mathcal{A}$ and $\mathcal{B}$ but we expect that these new sets will be of smaller cardinality than $L_{i_1},\cdots L_{i_k}$ and no longer contain vectors which are not the rows of some linear equivalence matrix.

Refinement process
Input: An integer k, two sets of polynomials $\mathcal{A}$, $\mathcal{B}$ and a set $\{L_1, \cdots, L_n\} \subset (\mathbb{F}_q^n)^n$.
Output: A set $\{\widetilde{L}_1, \cdots, \widetilde{L}_n\} \subset (\mathbb{F}_q^n)^n$.
Initialization: $\widetilde{L}_1 = \cdots = \widetilde{L}_n = \emptyset$ and $I = \{1, \cdots, n\}$
Choose $U \subseteq \mathbb{F}_q$
While $I \neq \emptyset$ **do**

Choose $\{i_1, \cdots, i_k\} \in I$ randomly

For $(l_{i_1}, \cdots, l_{i_k}) \in L_{i_1} \times \cdots L_{i_k}$ **do**

If $\mathcal{B}(\sum_{j=1}^{k} c_j l_{i_j}) = \mathcal{A}(\sum_{j=1}^{k} c_j e_{i_j}), \forall (c_1, \cdots, c_k) \in U \times \cdots \times U$ **then**
$\widetilde{L}_{i_1} \leftarrow \widetilde{L}_{i_1} \cup \{l_{i_1}\}, \cdots, \widetilde{L}_{i_k} \leftarrow \widetilde{L}_{i_k} \cup \{l_{i_k}\}$
EndIf

EndFor

$I \leftarrow I \setminus \{i_1, \cdots, i_k\}$

EndWhile
Return $\{\widetilde{L}_1, \cdots, \widetilde{L}_n\}$

The while loop can be divided into $\frac{n}{k}$ independent tasks, each of them consists of computing $\{\widetilde{L}_{i_1}, \cdots, \widetilde{L}_{i_k}\}$ from $\{\widetilde{L}_1, \cdots, \widetilde{L}_k\}$. Hence, this tasks can be independently computed on $\frac{n}{k}$ different processors.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399