On the computational complexity of some equivalence problems of polynomial systems of equations over finite fields

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

Abstract

We study in this paper the computational complexity of some equivalence relations on polynomial systems of equations over finite fields. These problems are analyzed with respect to polynomial-time many-one reductions (resp. Turing reductions, Levin reductions). In particular, we show that some of these problems are between P and NP. To do so, we compare these problems with the Graph Isomorphism problem. Moreover, using Interactive Proofs [9] we prove that, provided the Polynomial Hierarchy does not collapse, these problems are not NP-Complete (resp. NP-Hard).

1 Introduction

An interesting problem, both from a practical and theoretical point of view in computer science is the *Graph Isomorphism* problem (GI). Briefly, we recall that GI is the problem of deciding if there exists a bijective function p between two undirected graphs $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$ such that $(i,j) \in E_1$ if, and only if, $\big(p(i), p(j)\big) \in E_2$.

Using a group-theoretical approach of GI, Mathon has given the first indication that GI is likely not an NP-Complete problem [12]. Finally, Interactive Proof (IP) systems [9] and a result of Boppana, Hastad, and Zachos [1] have permitted to show that, provided the Polynomial Hierarchy (PH) does not collapse, GI is not NP-Complete.

Zero-knowledge (ZK) proofs are, loosely speaking, proofs that yield nothing beyond the validity of the assertion [7]. Such proofs, introduced by Goldwasser, Micali and Rackoff in [9], have permitted to give a rigorous framework for a typical problem in cryptography: authentication. Goldreich, Micali and Wigderson have presented a non-trivial example of ZK proof system based on the *Graph Isomorphism* problem (GI) [8]. Unfortunately an authentication protocol based on GI will not be practical at all. Indeed, algorithms for GI being very efficient [4], it will impose to use huge public keys (i.e. graphs) to achieve a reasonable level of security. To circumvent this problem, a variant of GI called *Isomorphism of Polynomials with one Secret* problem (IP1S), has been introduced by Patarin in [13]. It 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:

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

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 [13].

The main motivation of this paper is to study, from a complexity-theoretic point of view, the security of cryptographic schemes based on IP1S. To do so, we address here two relevant variants of it. The problem we call *Polynomial Linear Equivalence* problem (PLE) is the restriction of IP1S to bijective linear mappings. This is in fact not really a restriction since Perret proved, in [14], that IP1S and PLE are equivalent[1]. We also study the *Polynomial Isomorphism Equivalence* problem (PIE) which is the restriction of PLE to permutations.

1.1 Previous Work

To the best of our knowledge, PLE and PIE have not been, from a complexity-theoretic point of view, previously studied. However, we mention that, in [2], it is claimed that IP1S is at least as difficult as GI. Unfortunately, the least we can say is that the proof given in the extended version of that paper [3] is not clear and thus to our opinion subject to caution. We would like to emphasize that one of the aims of this paper is to give proofs of results which seem natural regarding the kind of problems studied. We mention that some techniques used in this paper are adapted - in the context of polynomial systems of equations - from ones for GI.

1.2 Organization of the paper and main results

This paper is organized as follows. In section 2, we introduce the notations and define more formally the problems studied.

Section 3 contains several properties that are used in the other sections. We present, in particular, some structural properties of PLE and PIE. We also show that some well known group-theoretical results in the Graph Isomorphism context can be extended to PIE and PLE.

Section 4 is devoted to the study of PIE, its decisional version called dPIE, and its counting version called #PIE. We give a lower bound on the theoretical complexity of[2] (d)PIE. To do so, we prove that GI is polynomial-time many-one reducible to dPIE. We then show that dPIE behaves differently than NP-Complete problems, by proving that it is Turing equivalent to its counting version #PIE. Finally, we conclude this section by proving that, provided the PH does not collapse, dPIE (resp. PIE) is not NP-Complete (resp. NP-Hard).

In section 5, we investigate more particularly dPLE and PLE. Similarly to (d)PIE, we give a lower bound on the complexity of (d)PLE, by proving that PIE is polynomial-time Turing reducible to dPLE. We also present a result permitting to improve the efficiency of cryptographic schemes based on PLE. Finally, we obtain, similarly to section 4 that, provided the PH does not collapse, dPLE (resp. PLE) is not NP-Complete (resp. NP-Hard).

[1] under Levin reductions

[2] (d)PIE is a shortcut for dPIE and PIE

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[3] (p a prime, and $r \geq 1$), by $\underline{x}$ the vector $(x_1, \ldots, x_n)$, and by $\mathbb{F}_q[\underline{x}] = \mathbb{F}_q[x_1, \ldots, x_n]$, the polynomial ring in the n indeterminates $x_1, \ldots, x_n$ over $\mathbb{F}_q$.

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 $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. Let $f \in \mathbb{F}_q[\underline{x}]$, the *leading term* of f denoted $LT(f)$, is the largest term among the terms of f with respect to some admissible ordering on the monomials, and 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, i.e. $f = \sum_d f^{(d)}$, 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. We extend this notation to vectors of polynomials. We shall denote by $\underline{a}^{(d)} = (a_1^{(d)}, \ldots, a_u^{(d)})$ the dth homogeneous components of the polynomials of $\underline{a} = \big(a_1, \cdots, a_u\big) \in \mathbb{F}_q[\underline{x}]^u$.

We shall denote by $\mathcal{S}_n$ the symmetric group on $\{1, \ldots, n\}$ and for $\sigma \in \mathcal{S}_n$ we set $\underline{x}\sigma = (x_{\sigma(1)}, \ldots, x_{\sigma(n)})$. We shall call $\mathcal{M}_{n,u}(\mathbb{F}_q)$ the set of $n \times u$ matrices with components in $\mathbb{F}_q$, $GL_n(\mathbb{F}_q)$ denotes as usual the set of invertible matrices of $\mathcal{M}_{n,n}(\mathbb{F}_q)$, and $AGL_n(\mathbb{F}_q)$ is the the cartesian product $GL_n(\mathbb{F}_q) \times \mathbb{F}_q^n$.

Finally, for the definitions of reductions used throughout this paper, namely polynomial-time many-one reduction (resp. Turing reduction, Levin reduction), the reader is referred to [5].

2.2 Polynomial equivalence problems

Let $\underline{a} = (a_1, \ldots, a_u)$ and $\underline{b} = (b_1, \ldots, b_u)$ be two sets of polynomials over $\mathbb{F}_q[\underline{x}]$. We mainly study in this paper two variants of IP1S. A first natural restriction is to consider linear bijective mappings. We shall say that $\underline{a}$ and $\underline{b}$ are *linear-equivalent*, denoted by $\underline{a} \equiv_L \underline{b}$, if there exists $S \in GL_n(\mathbb{F}_q)$, such that:

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

In the sequel we shall denote, for convenience, equations (1) by $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$.

We call such a matrix a *linear equivalence matrix*. The *Polynomial Linear Equivalence* problem (PLE) is then the one of finding - if any - a linear equivalence matrix between the two sets of polynomials $\underline{a}$ and $\underline{b}$. Moreover, we shall call dPLE the decisional version of PLE and L_{dPLE} denotes the language associated to dPLE. Note that, from a practical point of view, IP1S and PLE have been studied in [6], [11] (where IP1S is called Polynomial Affine Equivalence problem), [15], and [14].

We shall say that $\underline{a}$ and $\underline{b}$ are *isomorphic*, denoted by $\underline{a} \sim \underline{b}$, if there exists $\sigma \in \mathcal{S}_n$ such that:

$$\underline{b}(\underline{x}) = \underline{a}(\underline{x}\sigma).$$

[3] note that p, the characteristic of $\mathbb{F}_q$, is denoted char($\mathbb{F}_q$) in this paper

We call such a permutation an *equivalence permutation.* The *Polynomial Isomorphism Equivalence* (PIE) problem is then the one of finding - if any - an equivalence permutation between the two sets of polynomials $\underline{a}$ and $\underline{b}$. Moreover, we shall call dPIE the decisional version of PIE and L_{dPIE} denotes the language associated to dPLE. Finally, #PIE denotes the problem of counting the number of solutions of an instance of PIE; #PLE is defined similarly.

Remark 2.1. *In the sequel, we restrict our attention to instances* $(\underline{a},\underline{b})$ *of (d)PIE (resp. (d)PLE) such that given* $\sigma \in \mathcal{S}_n$ *(resp.* $S \in GL_n(\mathbb{F}_q)$*) the equality* $\underline{b}(\underline{x}) = \underline{a}(\underline{x}\sigma)$ *(resp.* $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$*) can be checked in polynomial-time. For (d)PIE, this simply implies that the polynomials of* $\underline{a}$ *and* $\underline{b}$ *have a polynomial (in* n*) number of terms. For (d)PLE, we restrict our attention to instances used in cryptographic applications, i.e. composed of polynomials of highest total degree equal to* 2 *or* 3*. Remember that the number of monomials of total degree* d *is equal to* $\binom{n+d}{n} \leq \frac{(n+d)^d}{d!} \leq (n+d)^d$*. Therefore, as soon as the highest total degree of the polynomials of* $\underline{a}$ *and* $\underline{b}$ *is fixed, the number of terms in* $\underline{a}(\underline{x}S)$ *is polynomial in* n*, and the equality* $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$ *can be checked in polynomial-time, i.e. dPLE* $\in NP$*.*

3 Properties of PLE and PIE

We present in this part several properties of PLE and PIE. We first investigate, in 3.1, structural properties of PLE and PIE. In 3.2, we show that some well known group-theoretical results in the Graph Isomorphism context can be extended to PLE and PIE. For a description of the group-theoretical approach of GI, we refer the reader to [10].

3.1 Structural Properties

For all $i, 1 \leq i \leq u$, D_i denotes the degree of the homogeneous component of highest degree in a_i. We stress that $\underline{a} \equiv_L \underline{b}$ implies that, for all $i, 1 \leq i \leq u$, a_i and b_i must have the same highest total degree D_i.

Proposition 3.1. *Let* $D = max_{1\leq i\leq u}(D_i)$ *and* $S \in GL_n(\mathbb{F}_q)$*. We have:*

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

Proof. Let $S \in GL_n(\mathbb{F}_q)$, such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$. For all $i, 1 \leq i \leq u$, 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 $S \in GL_n(\mathbb{F}_q)$ such that for all $i, 1 \leq i \leq u$, and for all $d, 0 \leq d \leq D$, $b_i^{(d)}(\underline{x}) = a_i^{(d)}(\underline{x}S)$. 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. $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$. □

Remark that proposition 3.1 already appeared, without proof, in [11]. It is quoted here for the sake of completeness.

Following a similar idea we obtain for PIE:

Proposition 3.2. *Let* $D = max_{1\leq i\leq u}(D_i)$ *and* $\sigma \in \mathcal{S}_n$*. We have:*

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

In some case, we can easily check that $\underline{a} \not\sim \underline{b}$. Indeed:

Lemma 3.1. *For all i, $1 \leq i \leq u$, let $|a_i|$ and $|b_i|$ be the number of terms of a_i and b_i. If there exist $i, 1 \leq i \leq u$ and $d, 1 \leq d \leq D$, such that $|a_i^{(d)}| \neq |b_i^{(d)}|$, then $\underline{a} \not\sim \underline{b}$.*

Proof. We prove in fact that if $\underline{b}(\underline{x}) = \underline{a}(\underline{x}\sigma)$, for some $\sigma \in \mathcal{S}_n$, then $|b_i| = |a_i|$, for all $i, 1 \leq i \leq u$. Remark that for all $i, 1 \leq i \leq u$, $|a_i(\underline{x}\sigma)| \leq |a_i(\underline{x})|$, i.e. the permutation σ do not increase the number of terms of $a_i(\underline{x}\sigma)$. Moreover, since $\underline{b}(\underline{x}\sigma^{-1}) = \underline{a}(\underline{x})$ also holds, we get that for all $i, 1 \leq i \leq u$, $|b_i(\underline{x})| = |a_i(\underline{x}\sigma)| \leq |a_i(\underline{x})|$ and $|a_i(\underline{x})| = |b_i(\underline{x}\sigma^{-1})| \leq |b_i(\underline{x})|$, i.e. $|b_i| = |a_i|$, for all $i, 1 \leq i \leq u$. □

Let $\underline{0_n}$ be the null vector of $\mathbb{F}_q^n$. For $d = 0$, we obtain that if $a_i(\underline{0_n}) \neq b_i(\underline{0_n})$, for some i, then $\underline{a} \not\sim \underline{b}$ (this last result also holds for PLE).

3.2 A group-theoretical approach of PLE and PIE

We first introduce the following notations. For $(\underline{a}, \underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$, we shall denote by $L_{(\underline{a},\underline{b})} = \{S \in GL_n(\mathbb{F}_q) : \underline{b}(\underline{x}) = \underline{a}(\underline{x}S)\}$, the set of linear equivalence matrices between $\underline{a}$ and $\underline{b}$. Moreover, we shall call:

$$\begin{aligned} Aut_{GL_n(\mathbb{F}_q)}(\underline{a}) &= \{S \in GL_n(\mathbb{F}_q) : \underline{a}(\underline{x}) = \underline{a}(\underline{x}S)\}, \text{ and} \\ Aut_{GL_n(\mathbb{F}_q)}(\underline{b}) &= \{S \in GL_n(\mathbb{F}_q) : \underline{b}(\underline{x}) = \underline{b}(\underline{x}S)\}, \end{aligned}$$

the *automorphism groups*, w.r.t. $GL_n(\mathbb{F}_q)$, of $\underline{a}$ and $\underline{b}$ respectively. One can see at once that $Aut_{GL_n(\mathbb{F}_q)}(\underline{a})$ and $Aut_{GL_n(\mathbb{F}_q)}(\underline{b})$ are subgroups of $GL_n(\mathbb{F}_q)$.

Proposition 3.3. *Let $S \in GL_n(\mathbb{F}_q)$, such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$. Then $L_{(\underline{a},\underline{b})}$ is a left (resp. right) coset, in $GL_n(\mathbb{F}_q)$, of the automorphism group $Aut_{GL_n(\mathbb{F}_q)}(\underline{b})$ (resp. $Aut_{GL_n(\mathbb{F}_q)}(\underline{a})$). That is $L_{(\underline{a},\underline{b})} = Aut_{GL_n(\mathbb{F}_q)}(\underline{b})S$ (resp. $L_{(\underline{a},\underline{b})} = SAut_{GL_n(\mathbb{F}_q)}(\underline{a})$).*

Proof. We first show that $L_{(\underline{a},\underline{b})} = Aut_{GL_n(\mathbb{F}_q)}(\underline{b})S$.
Let $S' \in L_{(\underline{a},\underline{b})}$, i.e. $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S')$. We have $\underline{b}(\underline{x}S^{-1}) = \underline{b}(\underline{x}S'^{-1})$, and so $\underline{b}(\underline{x}S'S^{-1}) = \underline{b}(\underline{x})$. Thus $S'S^{-1} \in Aut_{GL_n(\mathbb{F}_q)}(\underline{b})$, i.e. $S' \in Aut_{GL_n(\mathbb{F}_q)}(\underline{b})S$.
Now, let $M \in Aut_{GL_n(\mathbb{F}_q)}(\underline{b})S$, i.e. $M = S'S$ for some $S' \in Aut_{GL_n(\mathbb{F}_q)}(\underline{b})$. Thus, since $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, we get by definition of S' that $\underline{b}(\underline{x}) = \underline{b}(\underline{x}S') = \underline{a}(\underline{x}M)$, i.e. $M \in L_{(\underline{a},\underline{b})}$.
Similarly, one can prove that $L_{(\underline{a},\underline{b})} = SAut_{GL_n(\mathbb{F}_q)}(\underline{a})$. □

From this proposition, we deduce:

Corollary 3.1. *If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, for some $S \in GL_n(\mathbb{F}_q)$, then:*

$$|L_{(\underline{a},\underline{b})}| = |Aut_{GL_n(\mathbb{F}_q)}(\underline{b})| = |Aut_{GL_n(\mathbb{F}_q)}(\underline{a})|.$$

Finally:

Proposition 3.4. *Let $S \in GL_n(\mathbb{F}_q)$, such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, then $Aut_{GL_n(\mathbb{F}_q)}(\underline{a})$ and $Aut_{GL_n(\mathbb{F}_q)}(\underline{b})$ are conjugate, i.e. $Aut_{GL_n(\mathbb{F}_q)}(\underline{a}) = SAut_{GL_n(\mathbb{F}_q)}(\underline{b})S^{-1}$.*

Proof. We first remark that, for each $u \geq 1$, the linear group $GL_n(\mathbb{F}_q)$ acts on the $\mathbb{F}_q[\underline{x}]$-modulus $\mathbb{F}_q[\underline{x}]^u$, through the following map:

$$\begin{array}{rlcl} \phi_u : & GL_n(\mathbb{F}_q) \times \mathbb{F}_q[\underline{x}]^u & \rightarrow & \mathbb{F}_q[\underline{x}]^u \\ & \big(G, \underline{a}(\underline{x})\big) & \mapsto & a(\underline{x}S) \end{array}$$

In this context $Aut_{GL_n(\mathbb{F}_q)}(\underline{a})$ (resp. $Aut_{GL_n(\mathbb{F}_q)}(\underline{b})$) is also called *stabilizer* of $\underline{a}$ (resp. $\underline{b}$). Thus, since there exists $S \in GL_n(\mathbb{F}_q)$ such that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S)$, then $\underline{b}$ lies in the $GL_n(\mathbb{F}_q)$-orbit of $\underline{a}$. Thus, we immediately deduce that the stabilizers of $\underline{a}$ and $\underline{b}$ are conjugate and more precisely $Aut_{GL_n(\mathbb{F}_q)}(\underline{a}) = S Aut_{GL_n(\mathbb{F}_q)}(\underline{b})S^{-1}$. □

Let us consider now the Polynomial Isomorphism Equivalence problem.
Extending the notations given for PLE, $P_{(\underline{a},\underline{b})} = \{\sigma \in \mathcal{S}_n : \underline{b}(\underline{x}) = \underline{a}(\underline{x}\sigma)\}$ shall denote the set of equivalence permutations between $\underline{a}$ and $\underline{b}$. We shall also call $Aut_{\mathcal{S}_n}(\underline{a}) = \{\sigma \in \mathcal{S}_n : \underline{a}(\underline{x}) = \underline{a}(\underline{x}\sigma)\}$, and $Aut_{\mathcal{S}_n}(\underline{b}) = \{\sigma \in \mathcal{S}_n : \underline{b}(\underline{x}) = \underline{b}(\underline{x}\sigma)\}$, the *automorphism groups*, w.r.t. $\mathcal{S}_n$, of $\underline{a}$ and $\underline{b}$ respectively. We then have:

Proposition 3.5. *If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}\sigma)$, for some $\sigma \in \mathcal{S}_n$, then $P_{(\underline{a},\underline{b})}$ is a left (resp. right) coset, in $\mathcal{S}_n$, of the automorphism group $Aut_{\mathcal{S}_n}(\underline{b})$(resp. $Aut_{\mathcal{S}_n}(\underline{a})$). That is $P_{(\underline{a},\underline{b})} = Aut_{\mathcal{S}_n}(\underline{b})\sigma$ (resp. $P_{(\underline{a},\underline{b})} = \sigma Aut_{\mathcal{S}_n}(\underline{a})$).*

We then deduce:

Corollary 3.2. *If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}\sigma)$, for some $\sigma \in \mathcal{S}_n$, then:*

$$|P_{(\underline{a},\underline{b})}| = |Aut_{\mathcal{S}_n}(\underline{b})| = |Aut_{\mathcal{S}_n}(\underline{a})|.$$

Finally:

Proposition 3.6. *If $\underline{b}(\underline{x}) = \underline{a}(\underline{x}\sigma)$, for some $\sigma \in \mathcal{S}_n$, then $Aut_{\mathcal{S}_n}(\underline{a})$ and $Aut_{\mathcal{S}_n}(\underline{b})$ are conjugate, i.e. $Aut_{\mathcal{S}_n}(\underline{a}) = \sigma Aut_{\mathcal{S}_n}(\underline{b})\sigma^{-1}$.*

The proofs of the those results are omitted since they are very similar to the ones already given for PLE.

4 The Polynomial Isomorphism Equivalence problem

Let $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$ be two undirected graphs with the same set of vertices V, and with set of edges E_1 and E_2 respectively. We recall that the Graph Isomorphism problem is the one of deciding if there exists a bijective function $p : V \to V$ such that:

$$(i, j) \in E_1 \text{ if, and only if, } \big(p(i), p(j)\big) \in E_2.$$

We start this part by proving that GI is polynomial-time many-one reducible to dPIE. Remember that the $n \times n$ adjacency matrix $M_{\mathcal{G}}$ of a graph $\mathcal{G} = \big(V = \{1, \ldots, n\}, E\big)$ is defined for all $i, j, 1 \leq i, j \leq n$, by $\big(M_{\mathcal{G}}\big)_{i,j} = 1$ iff $(i, j) \in E$ and $\big(M_{\mathcal{G}}\big)_{i,j} = 0$ otherwise.

Proposition 4.1. GI$\leq_p^m$dPIE.

Proof. Let $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$ be two undirected graphs with the same set of vertices V, and with set of edges E_1 and E_2 respectively. Moreover, let $n = |V|$, and f be a mapping from the instances of GI to the instances of dPIE, defined such that:

$$f(\mathcal{G}_1, \mathcal{G}_2) = (a_2, a_1),$$

with $a_1(\underline{x}) = \underline{x}M_{\mathcal{G}_1}\underline{x}^t$ and $a_2(\underline{x}) = \underline{x}M_{\mathcal{G}_2}\underline{x}^t$ being polynomials of $\mathbb{F}_q[\underline{x}]$ (with char($\mathbb{F}_q$)$\neq 2$). We shall now prove that f is a polynomial-time many-one reduction from GI to dPIE.

Let $(\mathcal{G}_1, \mathcal{G}_2) \in L_{GI}$, i.e. there exists a bijective function $p : V \to V$ such that $(i,j) \in E_1$ if, and only if $\big(p(i), p(j)\big) \in E_2$. We then easily deduce that the adjacency matrices of $\mathcal{G}_1$ and $\mathcal{G}_2$ are such that:

$$\big(M_{\mathcal{G}_1}\big)_{i,j} = \big(M_{\mathcal{G}_2}\big)_{p(i),p(j)}, \text{ for all } i,j, 1 \leq i,j \leq n.$$

Thus, if we denote by P the permutation matrix associated to p (i.e. for all $i,j, 1 \leq i,j \leq n$, $P_{i,j} = 1$ iff $p(i) = j$ and $P_{i,j} = 0$ otherwise), then $M_{\mathcal{G}_1} = PM_{\mathcal{G}_2}P^t$ holds. Indeed, for all $i,j, 1 \leq i,j \leq n$, we have:

$$\begin{aligned}
(PM_{\mathcal{G}_2}P^t)_{i,j} &= \sum_{k=1}^{n} P_{i,k}\Big(\sum_{\ell=1}^{n}(M_{\mathcal{G}_2})_{k,\ell}P_{j,\ell}\Big) \\
&= \sum_{k=1}^{n} P_{i,k}\big((M_{\mathcal{G}_2})_{k,\ell_j}P_{j,\ell_j}\big) = \sum_{k=1}^{n} P_{i,k}(M_{\mathcal{G}_2})_{k,p(j)} \\
(PM_{\mathcal{G}_2}P^t)_{i,j} &= P_{i,k_i}(M_{\mathcal{G}_2})_{k_i,p(j)} = (M_{\mathcal{G}_2})_{p(i),p(j)} = (M_{\mathcal{G}_1})_{i,j}
\end{aligned}$$

Thus $a_1(\underline{x}) = \underline{x}M_{\mathcal{G}_1}\underline{x}^t = \underline{x}PM_{\mathcal{G}_2}P^t\underline{x}^t = a_2(\underline{x}P)$, i.e. $f(\mathcal{G}_1, \mathcal{G}_2) = (a_2, a_1) \in L_{dPIE}$.
Now, let $f(\mathcal{G}_1, \mathcal{G}_2) = (a_2, a_1) \in L_{dPIE}$, i.e. $a_1(\underline{x}) = a_2(\underline{x}\sigma)$, for some $\sigma \in \mathcal{S}_n$. Let also P_σ be the permutation matrix associated to σ (i.e. for all $i,j, 1 \leq i,j \leq n$, $\big(P_\sigma\big)_{i,j} = 1$ iff $\sigma(i) = j$ and $\big(P_\sigma\big)_{i,j} = 0$ otherwise). By construction, a_2 and a_1 are homogeneous polynomials of degree 2, thus there exist symmetric matrices M_1 and M_2 such that $a_1 = \underline{x}M_1\underline{x}^t$ and $a_2 = \underline{x}M_2\underline{x}^t$. Since char$(\mathbb{F}_q) \neq 2$, these matrices are unique, thus $M_1 = M_{\mathcal{G}_1}$ and $M_2 = M_{\mathcal{G}_2}$. Finally, $a_1(\underline{x}) = a_2(\underline{x}\sigma)$, implies that $\underline{x}M_{\mathcal{G}_1}\underline{x}^t = \underline{x}PM_{\mathcal{G}_2}P^t\underline{x}^t$ and thus $M_{\mathcal{G}_1} = P_\sigma M_{\mathcal{G}_2}P_\sigma^t$, i.e. $(\mathcal{G}_1, \mathcal{G}_2) \in L_{GI}$. □

Proposition 4.2. dPIE *is polynomial-time Turing equivalent to* PIE.

Proof. Obviously dPIE polynomial-time Turing reduces to PIE. Let us prove also that PIE polynomial-time Turing reduces to dPIE. The reduction is mainly based on the following remark.
Let $(\underline{a}, \underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$, and $\underline{a'_1} = (\underline{a}, x_1)$. Moreover, for all $j, 1 \leq j \leq n$, let $\underline{b'_j} = (\underline{b}, x_j)$. If for some $j_1, 1 \leq j_1 \leq n$, there exists an equivalence permutation σ between $\underline{a'_1}$ and $\underline{b'_{j_1}}$, we deduce that σ is an equivalence permutation between $\underline{a}$ and $\underline{b}$, transposing x_1 and x_{j_1}, i.e. $x_{j_1} = x_{\sigma(1)}$.
We can generalize this idea. Let $k, 1 \leq k \leq n$, and suppose that we have recovered for all $i, 1 \leq i \leq k-1$, x_{j_i} such that $x_{j_i} = x_{\sigma(i)}$ for some equivalence permutation σ between $\underline{a}$ and $\underline{b}$. In order to recover an equivalence permutation σ between $\underline{a}$ and $\underline{b}$ transposing x_k and x_{j_k}, we submit for all $j, 1 \leq j \leq n$, the polynomials $\underline{a'_k} = (\underline{a}, x_1, \ldots, x_k)$ and $\underline{b'_{k,j}} = (\underline{b}, x_{j_1}, \ldots, x_{j_{k-1}}, x_j)$ to a dPIE oracle. If for some $j_k, 1 \leq j_k \leq n$, there exists an equivalence permutation σ between $\underline{a'_k}$ and $\underline{b'_{k,j_k}}$, we deduce that σ is an equivalence permutation, between $\underline{a}$ and $\underline{b}$, transposing x_i and x_{j_i}, for all $i, 1 \leq i \leq k$.
Therefore for each i, $1 \leq i \leq n$, we recover x_{j_i} such that $x_{j_i} = x_{\sigma(i)}$ for some equivalence permutation σ between $\underline{a}$ and $\underline{b}$, with at most n queries to a dPIE oracle, i.e. one can recover a solution of PIE by sending at most n^2 queries to a dPIE oracle. □

Historically, the first indication that GI is not likely to be NP-Complete was given by Mathon [12]. He has shown that GI behaves differently than NP-Complete problems by proving that it is polynomial-time Turing equivalent to its counting version #GI. We present a similar result for dPIE. For this, we introduce the following problems. Given $\underline{a} = (a_1, \ldots, a_u) \in \mathbb{F}_q[\underline{x}]^u$, we shall call $\mathrm{PA}_{\mathcal{S}_n}$ the problem of finding - if any - $\sigma \in Aut_{\mathcal{S}_n}(\underline{a})$. Moreover, $\#\mathrm{PA}_{\mathcal{S}_n}$ is the counting problem associated to $\mathrm{PA}_{\mathcal{S}_n}$, i.e. the problem of finding the order of $Aut_{\mathcal{S}_n}(\underline{a})$.

Theorem 4.1. #PIE *is polynomial-time Turing equivalent to* dPIE.

Proof. One can see at once that dPIE is polynomial-time Turing reducible to #PIE.
To show that #PIE is Turing reducible to dPIE, we first remark that according to corollary 3.2, $\underline{a} \sim \underline{b}$ implies that $P_{\underline{a},\underline{b}} = |Aut_{\mathcal{S}_n}(\underline{a})| = |Aut_{\mathcal{S}_n}(\underline{b})|$. Thus #PIE is Turing reducible to $\#\mathrm{PA}_{\mathcal{S}_n}$. We now prove that $\#\mathrm{PA}_{\mathcal{S}_n}$ is Turing reducible to dPIE. The crucial point in this reduction is to remark that if for all $k, 1 \leq k \leq n$, $\underline{a'_k} = (\underline{a}, x_1, \ldots, x_k)$, we[4] then have:

$$|Aut_{\mathcal{S}_n}(\underline{a'_{k-1}})| = d_k |Aut_{\mathcal{S}_n}(\underline{a'_k})|, \tag{2}$$

with $d_k = |\{x_{\sigma(k)} : \sigma \in Aut_{\mathcal{S}_n}(\underline{a'_{k-1}})\}|$, i.e. the size of the orbit of x_k in $Aut_{\mathcal{S}_n}(\underline{a'_{k-1}})$.
To prove (2), we remark that for all $k, 1 \leq k \leq n$, $Aut_{\mathcal{S}_n}(\underline{a'_k})$ is a subgroup of $Aut_{\mathcal{S}_n}(\underline{a'_{k-1}})$. Thus, by Lagrange's theorem, we get that:

$$|Aut_{\mathcal{S}_n}(\underline{a'_{k-1}})| = \big(Aut_{\mathcal{S}_n}(\underline{a'_{k-1}}) : Aut_{\mathcal{S}_n}(\underline{a'_k})\big) |Aut_{\mathcal{S}_n}(\underline{a'_k})|.$$

Concluding the proof of (2) since $\big(Aut_{\mathcal{S}_n}(\underline{a'_{k-1}}) : Aut_{\mathcal{S}_n}(\underline{a'_k})\big)$, the index of $Aut_{\mathcal{S}_n}(\underline{a'_k})$ in $Aut_{\mathcal{S}_n}(\underline{a'_{k-1}})$ is equal to d_k. Therefore, since $|Aut_{\mathcal{S}_n}(\underline{a'_n})| = 1$, we get that:

$$|Aut_{\mathcal{S}_n}(\underline{a'_0})| = |Aut_{\mathcal{S}_n}(\underline{a})| = \prod_{k=1}^{n} d_k.$$

Moreover, for each $k, 1 \leq k \leq n$, d_k can be computed, by using a technique similar to the one described in the proof of proposition 4.2, by sending $n - k + 1$ queries to a dPIE oracle. Thus, the order of $Aut_{\mathcal{S}_n}(\underline{a})$ can be computed with at most n^2 queries to a dPIE oracle, i.e. $\#\mathrm{PA}_{\mathcal{S}_n}$ Turing reduces to dPIE. Therefore, #PIE is Turing equivalent to dPIE. □

Since PIE and dPIE are Turing equivalent, we get:

Corollary 4.1. #PIE *is Turing equivalent to* PIE.

In order to prove that dPIE is not NP-Complete, we introduce now Interactive Proof (IP) systems.

Definition 4.1. *An* Interactive Proof (IP) system *for a language L consists of a randomized polynomial-time algorithm called verifier, denoted by V, and a prover, denoted by P, which can make arbitrary many computations. The two players interact by sending messages to each other. After at most a polynomial number of communications, the verifier finally has to accept or reject a given input such that the following conditions hold:*

Completeness. *For all $x \in L$:*

$$\Pr[(V, P)(x) \ \textit{accepts}\ ] = 1.$$

[4] For $k = 0$, we set $\underline{a'_0} = \underline{a}$

Soundness. *For all $x \notin L$ and for any prover P^*:*

$$\Pr[(V,P^*)(x) \text{ accepts }] \leq \frac{1}{2}.$$

The probabilities are taken over the random choices of the verifier.
IP *denotes the set of languages having an interactive proof system. Finally,* IP$[k]$ *is a subset of* IP *denoting the set of languages having an interactive proof system where the prover and the verifier exchange at most k messages.*

In order to prove that dPIE is not NP-Complete, we will show that its complementary problem, denoted $\overline{\text{dPIE}}$, has a constant round Interactive Proof protocol.

Proposition 4.3. $\overline{\text{dPIE}} \in \text{IP}[2]$.

Proof. Let us prove that the following IP protocol accept $\overline{\text{dPIE}}$.

Input: $(\underline{a_0}, \underline{a_1}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$.
Question: Is $\underline{a_0} \not\sim \underline{a_1}$?
Protocol.
The Verifier picks $i \in \{0,1\}$, and $\sigma \in \mathcal{S}_n$, both uniformly at random, and sends $\underline{a'} = \underline{a_i}(\underline{x}\sigma)$ to the Prover.
The Prover answers by sending $j \in \{0,1\}$ to the Verifier.
The Verifier accepts if $i = j$ and rejects otherwise.

Completeness. If $\underline{a_0} \not\sim \underline{a_1}$, then a Prover can always determine whether $\underline{a'} \sim \underline{a_0}$ or $\underline{a'} \sim \underline{a_1}$. Thus the Verifier accepts with probability one.

Soundness. On the other hand, if $\underline{a_0} \sim \underline{a_1}$, then $\underline{a'}$ is isomorphic to both $\underline{a_0}$ and $\underline{a_1}$. In this case, we prove that $\underline{a'} = \underline{a_i}(\underline{x}\sigma)$ gives no information about i, i.e. the distribution of i given $\underline{a'}$ is equal to the distribution of i (which is chosen uniformly at random in $\{0,1\}$). We introduce for this a random variable ψ uniformly distributed over $\{0,1\}$ and a random variable Σ uniformly distributed over $\mathcal{S}_n$. We get:

Lemma 4.1. *Let $(\underline{a_0}, \underline{a_1}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$ such that $\underline{a_0} \sim \underline{a_1}$. Then, for all $\underline{a'} \in \mathbb{F}_q[\underline{x}]^u$, such that $\underline{a'} \sim \underline{a_0}$, we have $Pr[\psi = 0 \,|\, \underline{a_\psi}(\underline{x}\Sigma) = \underline{a'}] = Pr[\psi = 1 \,|\, \underline{a_\psi}(\underline{x}\Sigma) = \underline{a'}] = \frac{1}{2}$.*

Proof. Remark that $\Pr[\underline{a_\psi}(\underline{x}\Sigma) = \underline{a'} \,|\, \psi = 0] = \Pr[\underline{a_0}(\underline{x}\Sigma) = \underline{a'}] = \Pr[\Sigma \in L_{(\underline{a'},\underline{a_0})}]$. Moreover, according to corollary 3.2, we have $|P_{(\underline{a'},\underline{a_0})}| = |Aut_{\mathcal{S}_n}(\underline{a'})| = |P_{(\underline{a'},\underline{a_1})}|$, and thus $\Pr[\underline{a_0}(\underline{x}\Sigma) = \underline{a'}] = \Pr[\underline{a_1}(\underline{x}\Sigma) = \underline{a'}]$. Therefore, $\Pr[\underline{a_\psi}(\underline{x}\Sigma) = \underline{a'} \,|\, \psi = 0] = \Pr[\underline{a_\psi}(\underline{x}\Sigma) = \underline{a'} \,|\, \psi = 1]$. Concluding the proof, according to the Bayes formula. □

From this lemma, it follows that no prover (no matter what its strategy is) can guess i with probability greater than $\frac{1}{2}$. Therefore no Prover can fool the Verifier, into accepting that $\underline{a_0} \not\sim \underline{a_1}$, with probability greater than $\frac{1}{2}$. □

Finally, according to [1], we deduce an upper bound for the complexity of (d)PIE.

Corollary 4.2. *If* dPIE *(resp.* PIE*) is NP-Complete (resp. NP-Hard) then the Polynomial Hierarchy (PH) collapse at the second level, i.e. PH$= \Sigma_2^p$.*

5 The Polynomial Linear Equivalence problem

We investigate in this part (d)PLE.
The following result permits to give a lower bound on the theoretical complexity of dPLE.

Proposition 5.1. PIE *is polynomial-time Turing reducible to* dPLE.

Proof. The proof is similar to the one given in proposition 4.2, and the reduction is mainly based on the following remark.
Let $(\underline{a},\underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$, and $\underline{a'_1} = (\underline{a}, x_1)$. Moreover, for all $j, 1 \leq j \leq n$, let $\underline{b'_j} = (\underline{b}, x_j)$. If for some $j_1, 1 \leq j_1 \leq n$, there exists a linear equivalence matrix S between $\underline{a'_1}$ and $\underline{b'_{j_1}}$, we deduce that S is a linear equivalence between $\underline{a}$ and $\underline{b}$, transposing x_1 and x_{j_1}, i.e. for all $j, 1 \leq j \leq n$, $S_{1,j} = 1$ iff $j = j_1$ and $S_{1,j_1} = 0$ otherwise. Therefore, if σ is a permutation of $\mathcal{S}_n$ such that $x_{j_1} = x_{\sigma(1)}$, then we have:

$$\underline{b}(\underline{x}) = \underline{a}\big(x_{\sigma(1)}, S_2(\underline{x}), \ldots, S_n(\underline{x})\big),$$

with $\big(S_1(\underline{x}), \ldots, S_n(\underline{x})\big) = \underline{x}S$.
More generally, let $k, 1 \leq k \leq n$, and suppose that we have constructed $\sigma \in \mathcal{S}_n$ such that $x_{j_i} = x_{\sigma(i)}$, for all $i, 1 \leq i \leq k-1$, and:

$$\underline{b}(\underline{x}) = \underline{a}\big(x_{\sigma(1)}, \ldots, x_{\sigma(k-1)}, S_k(\underline{x}), \ldots, S_n(\underline{x})\big),$$

for some $S \in L_{(\underline{a},\underline{b})}$.
By submitting for all $j, 1 \leq j \leq n$, $\underline{a'_k} = (\underline{a}, x_1, \ldots, x_k)$ and $\underline{b'_{k,j}} = (\underline{b}, x_{j_1}, \ldots, x_{j_{k-1}}, x_j)$ to a dPLE oracle, one can recover - if any - $j_k, 1 \leq j_k \leq n$ such that there is a linear equivalence matrix S between $\underline{a'_k}$ and $\underline{b'_{k,j_k}}$. (We mention that if for all $j, \underline{a'_k} \not\equiv_L \underline{b'_{k,j}}$ then $\underline{a} \not\sim \underline{b}$). Thus, if we impose that σ also verifies $x_{j_k} = x_{\sigma(k)}$, then we have:

$$\underline{b}(\underline{x}) = \underline{a}\big(x_{\sigma(1)}, \ldots, x_{\sigma(k)}, S_{k+1}(\underline{x}), \ldots, S_n(\underline{x})\big).$$

Therefore one can see that a solution of PIE can be constructed by sending at most n^2 queries to a dPLE oracle. □

Since GI$\leq_p^m$dPIE and dPLE obviously polynomial-time Turing reduces to PLE, it holds that:

Corollary 5.1. GI *is polynomial-time Turing reducible to* dPLE *and* PLE.

We stress that proving GI$\leq_p^m$dPLE, as claimed in [3], seems hard to achieve.

We introduce now some new problems. We shall call Hom_PLE the restriction of PLE to homogeneous instances, i.e. each polynomial of $\underline{a}$ (resp. $\underline{b}$) is homogeneous. Moreover, dHom_PLE denotes the decisional version of Hom_PLE.

Proposition 5.2. dPLE *is polynomial-time many-one equivalent to* dHom_PLE.

Proof. One can see at once that the function $f(\underline{a},\underline{b}) = (\underline{a},\underline{b})$, for all $(\underline{a},\underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$, is a polynomial-time many-one reduction between dHom_PLE and dPLE. Thus dHom_PLE$\leq_p^m$dPLE.
Now, let $(\underline{a},\underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$ be an instance of dPLE and $D = max_{1 \leq i \leq u}(D_i)$. In order to prove that dPLE$\leq_p^m$dHom_PLE, we define a function $g : \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u \rightarrow \mathbb{F}_q[\underline{x}]^{(D+1)\cdot u} \times \mathbb{F}_q[\underline{x}]^{(D+1)\cdot u}$ in the following way. For all $(\underline{a},\underline{b}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$:

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

with $\underline{A} = \big(\underline{a}^{(D)}, \underline{a}^{(D-1)}, \ldots, \underline{a}^{(0)}\big)$ and $\underline{B} = \big(\underline{b}^{(D)}, \underline{b}^{(D-1)}, \ldots, \underline{b}^{(0)}\big)$.

Remark that by its very construction, $g(\underline{a}, \underline{b})$ is an instance of dHom_PLE.

Now, 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 proposition 3.1, we have $\underline{b}^{(d)}(\underline{x}) = \underline{a}^{(d)}(\underline{x}S)$, for all $d, 0 \leq d \leq D$. Thus $\underline{B}(\underline{x}) = \underline{A}(\underline{x}S)$, i.e. $g(\underline{a}, \underline{b}) = (\underline{A}, \underline{B}) \in L_{dHom\_PLE}$.

Let $g\big(\underline{a}, \underline{b}\big) = \big(\underline{A}, \underline{B}\big) \in L_{dHom\_PLE}$, i.e. $\underline{B}(\underline{x}) = \underline{A}(\underline{x}S')$, for some $S' \in GL_n(\mathbb{F}_q)$. Since $\underline{B}(\underline{x}) = \underline{A}(\underline{x}S')$ implies that $\underline{b}^{(d)}(\underline{x}) = \underline{a}^{(d)}(\underline{x}S')$, for all $d, 0 \leq d \leq D$, we then get by proposition 3.1 that $\underline{b}(\underline{x}) = \underline{a}(\underline{x}S')$. Thus $(\underline{a}, \underline{b}) \in L_{dPLE}$. □

Note that, according to proposition 3.2, this result also holds for dPIE and dHom_PIE, using an obvious notation.

Corollary 5.2. PLE *is polynomial-time Levin equivalent to* Hom_PLE.

Proof. Let f and g be defined as in the proof of proposition 5.2, and h be the identity mapping of $GL_n(\mathbb{F}_q)$. It is easy to see that (f, h, h) (resp. (g, h, h)) is a Levin reduction between Hom_PLE and PLE (resp. PLE and Hom_PLE). □

We stress that this result is also of practical interest. Indeed, for the same value of u, n and q, an instance of Hom_PLE have in general less terms than an instance of PLE. Thus, restricting PLE to Hom_PLE permits to decrease the size of the public key in cryptographic schemes based on PLE. Moreover, for the authentication scheme (resp. signature scheme) based on PLE, such a restriction also permits to decrease the size of the interactions between a Prover and a Verifier (resp. to decrease the size of a signature), without changing the theoretical complexity.

We show now that the complementary problem of dPLE, denoted $\overline{\text{dPLE}}$, has a constant round IP protocol. The proof is adapted to the one previously given for $\overline{\text{dPIE}}$. The only difference here is that the Verifier must randomly generate a matrix in $GL_n(\mathbb{F}_q)$.

We can proceed as follows, the verifier randomly generates a matrix in $\mathcal{M}_{n,n}(\mathbb{F}_q)$. If this matrix is invertible then the protocol is similar to the one described for dPIE, otherwise if (after several attempts) the Verifier fails to obtain an invertible matrix it accepts directly. The number of attempts is derived from this lemma:

Lemma 5.1. *We have:*

$$\frac{|GL_n(\mathbb{F}_q)|}{|\mathcal{M}_{n,n}(\mathbb{F}_q)|} \geq \frac{1}{4}.$$

Proof. Since $\frac{|GL_n(\mathbb{F}_q)|}{|\mathcal{M}_{n,n}(\mathbb{F}_q)|} = \prod_{k=1}^{n}(1 - \frac{1}{q^k})$, we have $\frac{|GL_n(\mathbb{F}_q)|}{|\mathcal{M}_{n,n}(\mathbb{F}_q)|} \geq \frac{|GL_n(\mathbb{F}_2)|}{|\mathcal{M}_{n,n}(\mathbb{F}_2)|}$, concluding the proof, since at least $1/4$ of the matrices in $\mathcal{M}_{n,n}(\mathbb{F}_2)$ are invertible (see [16], p. 45). □

We can now describe the protocol:

Proposition 5.3. $\overline{\text{dPLE}} \in \text{IP}[2]$.

Proof. Consider the following IP protocol for $\overline{\text{dPLE}}$.

Input: $(\underline{a_0}, \underline{a_1}) \in \mathbb{F}_q[\underline{x}]^u \times \mathbb{F}_q[\underline{x}]^u$.
Question: Is $\underline{a_0} \not\equiv_L \underline{a_1}$?
Protocol.
The Verifier picks uniformly at random $i \in \{0,1\}$, and randomly generates matrices in $\mathcal{M}_{n,n}(\mathbb{F}_q)$. Until one matrix, say S, is invertible. The Verifier then sends $\underline{a'} = \underline{a_i}(\underline{x}S)$ to the Prover.
If after five trials no matrix is invertible, the Verifier stops and accepts directly.
The Prover answers by sending $j \in \{0,1\}$ to the Verifier.
The Verifier accepts if $i = j$ and rejects otherwise.

Completeness. Clearly, if $\underline{a_0} \not\equiv_L \underline{a_1}$ then the Prover will never fail in convincing the Verifier.

Soundness. If $\underline{a_0} \equiv_L \underline{a_1}$, then $\underline{a'}$ is linear equivalent to $\underline{a_0}$ and $\underline{a_1}$. Similarly to the proof of proposition 4.3, one can prove that $\underline{a'} = \underline{a_i}(\underline{x}S)$ gives no information about i, i.e. the distribution of i given $\underline{a'}$ is equal to the distribution of i (which is chosen uniformly at random in $\{0,1\}$). This is mainly due to the fact that, according to corollary 3.1, we have $|L_{(\underline{a'},\underline{a_0})}| = |Aut_{GL_n}(\underline{a'})| = |L_{(\underline{a'},\underline{a_1})}|$.
It follows that no prover (no matter what its strategy is) can guess i with probability greater than $\frac{1}{2} + \left(\frac{3}{4}\right)^5 < \frac{9}{16}$ (with $\left(\frac{3}{4}\right)^5 < \frac{1}{16}$ being the probability of not obtaining an invertible matrix). Finally, by repeating the protocol two times, we obtain that no Prover can fool the Verifier into accepting that $\underline{a_0} \not\equiv_L \underline{a_1}$ with probability greater than $\left(\frac{9}{16}\right)^2 < \frac{1}{2}$. □
We then obtain:

Corollary 5.3. *If* dPLE *(resp.* PLE*) is NP-Complete (resp. NP-Hard) then the Polynomial Hierarchy (PH) collapse at the second level, i.e. PH=* Σ_2^p.

6 Conclusion

From a complexity-theoretic point of view, we have proved that dPIE (resp. dPLE) is at least as difficult as the Graph Isomorphism problem. It also appears that, using Interactive Proofs, dPIE and dPLE (resp. PIE and PLE) are, assuming PH does not collapse, not NP-Complete (resp. NP-Hard).
We also believe that the group-theoretical properties given in 3.2 could be used in the design of an (efficient) algorithm for PLE or PIE.

References

[1] R. B. Boppana, J. Hastad, and S. Zachos,“Does co-NP have short interactive proofs?”, *Inform. Process. Lett.*, 25(2):127–132, 1987.

[2] 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.

[3] N. Courtois, L. Goubin, and J. Patarin, “Improved Algorithms for Isomorphism of Polynomials-Extended Version”. Available from `www.minrank.org/ip6long.ps`

[4] S. Fortin,“The graph isomorphism problem”. Technical Report 96-20, University of Alberta, 1996.

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

[6] W.Geiselmann, W.Meier, and R.Steinwandt, "An Attack on the Isomorphisms of Polynomials Problem with One Secret," *Int. Journal of Information Security*, 2(1): pp. 59-64, 2003.

[7] O.Goldreich, "FOUNDATIONS OF CRYPTOGRAPHY, Basic Tools." Cambrige University press, 392 pp.

[8] 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.

[9] S. Goldwasser, S. Micali and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems". SIAM Journal on Computing, Vol. 18, pp. 186–208, 1989.

[10] M. Hoffman, "Group-theoretic algorithms and graph isomorphism", *Lecture Notes in Computer Science*,vol. 136, Springer-Verlag, 982.

[11] 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.

[12] R. Mathon, "A note on the graph isomorphism counting problem", *Inform. Process. Lett.* 8, pp. 131–132, 1979.

[13] 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.

[14] L. Perret, "A fast cryptanalysis of the Isomorphism of Polynomials with one Secret problem", submitted.

[15] L. Perret, and A. Bayad, "A differential approach to a polynomial equivalence problem", in Proceedings of ISIT 2004, extended abstract, pp. 142, 2004.

[16] T.Thierauf, "The Computational Complexity of Equivalence and Isomorphism Problems", *Lecture Notes in Computer Science,* vol. 1852, Springer-Verlag, 2000.