Polynomial-Time Algorithms for Quadratic Isomorphism of Polynomials: The Regular Case

Jérémy Berthomieu, Jean-Charles Faugère, Ludovic Perret

Polynomial-Time Algorithms for Quadratic Isomorphism of Polynomials: The Regular Case

Jérémy Berthomieu[a,b,c,*], Jean-Charles Faugère[c,a,b], Ludovic Perret[a,b,c]

[a] *Sorbonne Universités,* UPMC *Univ Paris 06, Équipe* POLSYS, LIP6*, F-75005, Paris, France*
[b] CNRS*,* UMR 7606*,* LIP6*, F-75005, Paris, France*
[c] INRIA*, Équipe* POLSYS*, Centre Paris – Rocquencourt, F-75005, Paris, France*

Abstract

Let $\mathbf{f} = (f_1, \ldots, f_m)$ and $\mathbf{g} = (g_1, \ldots, g_m)$ be two sets of $m \geq 1$ nonlinear polynomials in $\mathbb{K}[x_1, \ldots, x_n]$ ($\mathbb{K}$ being a field). We consider the computational problem of finding – if any – an invertible transformation on the variables mapping $\mathbf{f}$ to $\mathbf{g}$. The corresponding equivalence problem is known as *Isomorphism of Polynomials with one Secret* (IP1S) and is a fundamental problem in multivariate cryptography. Amongst its applications, we can cite Graph Isomorphism (GI) which reduces to equivalence of cubic polynomials with respect to an invertible linear change of variables, according to Agrawal and Saxena. The main result is a randomized polynomial-time algorithm for solving IP1S for quadratic instances – a particular case of importance in cryptography.

To this end, we show that IP1S for quadratic polynomials can be reduced to a variant of the classical module isomorphism problem in representation theory. We show that we can essentially *linearize* the problem by reducing quadratic-IP1S to test the orthogonal simultaneous similarity of symmetric matrices; this latter problem was shown by Chistov, Ivanyos and Karpinski (ISSAC 1997) to be equivalent to finding an invertible matrix in the linear space $\mathbb{K}^{n\times n}$ of $n \times n$ matrices over $\mathbb{K}$ and to compute the square root in a certain representation in a matrix algebra. While computing square roots of matrices can be done efficiently using numerical methods, it seems difficult to control the bit complexity of such methods. However, we present exact and polynomial-time algorithms for computing a representation of the square root of a matrix in $\mathbb{K}^{n\times n}$, for various fields (including finite fields), as a product of two matrices. Each coefficient of these matrices lie in an extension field of $\mathbb{K}$ of polynomial degree. We then consider #IP1S, the counting version of IP1S for quadratic instances. In particular, we provide a (complete) characterization of the automorphism group of homogeneous quadratic polynomials. Finally, we also consider the more general *Isomorphism of Polynomials* (IP) problem where we allow an invertible linear transformation on the variables *and* on the set of polynomials. A randomized polynomial-time algorithm for solv-

*Laboratoire d'Informatique de Paris 6, Université Pierre-et-Marie-Curie, Boîte Courrier 169, 4 place Jussieu, F-75252 Paris Cedex 05, France.

Email addresses: jeremy.berthomieu@lip6.fr (Jérémy Berthomieu), jean-charles.faugere@inria.fr (Jean-Charles Faugère), ludovic.perret@lip6.fr (Ludovic Perret)

ing IP when $\mathbf{f} = (x_1^d, \ldots, x_n^d)$ is presented. From an algorithmic point of view, the problem boils down to factoring the determinant of a linear matrix (*i.e.* a matrix whose components are linear polynomials). This extends to IP a result of Kayal obtained for PolyProj.

Keywords: Quadratic forms, computer algebra, polynomial isomorphism, multivariate cryptography, module isomorphism
2010 MSC: 12Y05, 94A60, 68W20, 68W30, 68Q25

1. Introduction

A fundamental question in computer science is to provide algorithms allowing to test if two given objects are *equivalent* with respect to some transformation. In this paper, we consider equivalence of nonlinear polynomials in several variables. Equivalence of polynomials has profound connections with a rich variety of fundamental problems in computer science, ranging – among others topics – from cryptography (*e.g.* Patarin (1996); Tang and Xu (2012, 2014); Yang et al. (2011)), arithmetic complexity (*via* Geometric Complexity Theory (GCT) for instance, see Bürgisser (2012); Kayal (2012); Mulmuley (2012); Mulmuley and Sohoni (2001)), testing low degree affine-invariant properties (Bhattacharyya et al. (2013); Green and Tao (2009); Grigorescu et al. (2013), ...). As we will see, the notion of equivalence can come with different flavours that impact the intrinsic hardness of the problem considered.

In Agrawal and Saxena (2006); Saxena (2006), the authors show that Graph Isomorphism reduces to equivalence of cubic polynomials with respect to an invertible linear change of variables (a similar reduction holds between $\mathbb{F}$-algebra Isomorphism and cubic equivalence of polynomials). This strongly suggests that solving equivalence problems efficiently is a very challenging algorithmic task.

In cryptography, the hardness of deciding equivalence between two sets of m polynomials with respect to an invertible linear change of variables is the security core of several cryptographic schemes: the seminal zero-knowledge ID scheme of Patarin (1996), and more recently group/proxy signature schemes (Tang and Xu (2012, 2014); Yang et al. (2011)). Note that there is a subtle difference between the equivalence problem considered in Agrawal and Saxena (2006); Kayal (2011); Saxena (2006) and the one considered in cryptographic applications.

Whilst Agrawal and Saxena (2006); Kayal (2011); Saxena (2006) restrict their attention to $m = 1$, arbitrary $m \geq 1$ is usually considered in cryptographic applications. In the former case, the problem is called *Polynomial Equivalence* (PolyEquiv), whereas it is called *Isomorphism of Polynomials with One Secret* (IP1S) problem in the latter case. We emphasize that the hardness of equivalence can drastically vary in function of m. An interesting example is the case of quadratic forms. The problem is completely solved when $m = 1$, but no polynomial-time algorithm exists for deciding simultaneous equivalence of quadratic forms. In this paper, we make a step ahead to close this gap by presenting a randomized polynomial-time algorithm for solving simultaneous equivalence of quadratic forms over various fields.

Equivalence of multivariate polynomials is also a fundamental problem in Multivariate Public-Key Cryptography (MPKC). This is a family of asymmetric (encryption and signature) schemes whose public-key is given by a set of m multivariate equations (Matsumoto and Imai (1988); Patarin (1996)). To minimize the public-key storage, the multivariate polynomials considered are usually quadratic. The basic idea of MPKC is to construct a public-key which is equivalent to a set of quadratic multivariate polynomials with a specific structure (see for instance Wolf and Preneel (2011)). Note that the notion of equivalence considered in this context is more general than the one considered for PolyEquiv or IP1S. Indeed, the equivalence is induced by an invertible linear change of variables and an invertible linear combination on the polynomials. The corresponding equivalence problem is known (Patarin (1996)) as *Isomorphism of Polynomials* (IP or IP2S).

PolyEquiv, IP, and IP1S are not NP-Hard unless the polynomial-hierarchy collapses, Perret (2004); Patarin et al. (1998). However, the situation changes drastically when considering the equivalence for more general linear transformations (in particular, not necessarily invertible). In this context, the problem is called PolyProj. At SODA'11, Kayal (2011) showed that PolyProj is NP-Hard. This may be due to the fact that various fundamental questions in arithmetic complexity can be re-interpreted as particular instances of PolyProj (see Bürgisser (2012); Kayal (2012); Mulmuley (2012); Mulmuley and Sohoni (2001)).

Typically, the famous VP vs VNP question (Valiant (1979)) can be formulated as an equivalence problem between the determinant and permanent polynomials. Such a link is in fact the core motivation of Geometric Complexity Theory. The problem of computing the symmetric rank (Bernardi et al. (2011); Comon et al. (2008)) of a symmetric tensor also reduces to an equivalence problem involving a particular multivariate polynomial (Kayal (2012)). To mention another fundamental problem, the task of minimizing the cost of computing matrix multiplication reduces to a particular equivalence problem (Bürgisser and Ikenmeyer (2011, 2013); Cohn and Umans (2013); Kayal (2012)).

Organization of the Paper and Main Results

Let $\mathbb{K}$ be a field, $\mathbf{f}$ and $\mathbf{g}$ be two sets of m polynomials over $\mathbb{K}[x_1,\ldots,x_n]$. The Isomorphism of Polynomials (IP) problem, introduced by Patarin (Patarin (1996)), is as follows:

Isomorphism of Polynomials (IP)

Input: $\big((\mathbf{f}=(f_1,\ldots,f_m) \text{ and } \mathbf{g}=(g_1,\ldots,g_m)\big) \in \mathbb{K}[x_1,\ldots,x_n]^m \times \mathbb{K}[x_1,\ldots,x_n]^m$.

Question: Find – if any – $(A,B) \in \mathrm{GL}_n(\mathbb{K}) \times \mathrm{GL}_m(\mathbb{K})$ such that:

$$\mathbf{g}(\mathbf{x}) = B \cdot \mathbf{f}(A \cdot \mathbf{x}), \text{ with } \mathbf{x} = (x_1,\ldots,x_n)^{\mathrm{T}}.$$

While IP is a fundamental problem in multivariate cryptography, there are quite few algorithms, such as Patarin et al. (1998); Bouillaguet et al. (2013); Faugère and Perret (2006), solving IP. In particular, Faugère and Perret (2006) proposed to solve IP by reducing it to a system of nonlinear equations whose variables are the unknown coefficients of the matrices. It was conjectured in Faugère and Perret (2006), but never proved, that the corresponding system of nonlinear equations can be solved in polynomial time as soon as the IP instances considered are not homogeneous.

Indeed, by slicing of the polynomials degree by degree, one can find equations in the coefficients of the transformation allowing one to recover the transformation. More recently, Bouillaguet et al. (2013) presented exponential (in the number of variables n) algorithms for solving quadratic homogeneous instances of IP over finite fields. This situation is clearly unsatisfying, and suggests that an important open problem for IP is to identify large class of instances which can be solved in (randomized) polynomial time.

An important special case of IP is the *IP problem with one secret* (IP1S for short), where B is the identity matrix. From a cryptographic point of view, the most natural case encountered for equivalence problems is inhomogeneous polynomials with affine transformations. For IP1S, we show that such a case can be handled in the same way as homogeneous instances with linear transformations (see Proposition 5). As a side remark, we mention that there exist more efficient methods to handle the affine case; typically by considering the homogeneous components, see Faugère and Perret (2006). However, homogenizing the instances allows us to make the proofs simpler and cleaner. As such, we focus our attention to solve IP1S for quadratic homogeneous forms.

When $m = 1$, the IP1S problem can be easily solved by computing a reduced form of the input quadratic forms. In Bouillaguet et al. (2011), the authors present an efficient heuristic algorithm for solving IP1S on quadratic instances. However, the algorithm requires to compute a Gröbner basis. So, its complexity could be exponential in the worst case. More recently, Macario-Rat et al. (2013) proposed a polynomial-time algorithm for solving IP1S on quadratic instances with $m = 2$ over fields of any characteristic. We consider here arbitrary $m > 1$.

In computer algebra, a fundamental and related problem is the simplification of a homogeneous polynomial system $\mathbf{f} \in \mathbb{K}[\mathbf{x}]^m$. That is, compute $A \in \mathrm{GL}_n(\mathbb{K})$ such that $\mathbf{g}(\mathbf{x}) = \mathbf{f}(A \cdot \mathbf{x})$ is easier to solve. In this setting, RIDGE algorithm (see Berthomieu et al. (2010); Hironaka (1970); Giraud (1972)) and MINVAR algorithm (see Carlini (2005); Kayal (2011)) reduce to the best the number of variables of the system. More generally, for a given homogeneous polynomial system $\mathbf{f}$, the *Functional Decomposition Problem* is the problem of computing $\mathbf{h} = (h_1, \ldots, h_s)$ homogeneous and $\mathbf{g}$ such that $\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{h}(\mathbf{x}))$.

To simplify the presentation in this introduction, we mainly deal with fields of characteristic $\neq 2$. Results for fields of characteristic 2 are also given later in this paper. Now, we define formally IP1S:

Definition 1. Let $\big(\mathbf{f} = (f_1, \ldots, f_m), \mathbf{g} = (g_1, \ldots, g_m)\big) \in \mathbb{K}[x_1, \ldots, x_n]^m \times \mathbb{K}[x_1, \ldots, x_n]^m$. We shall say that $\mathbf{f}$ and $\mathbf{g}$ are equivalent, denoted $\mathbf{f} \sim \mathbf{g}$, if $\exists\, A \in \mathrm{GL}_n(\mathbb{K})$ such that:

$$\mathbf{g}(\mathbf{x}) = \mathbf{f}(A \cdot \mathbf{x}).$$

IP1S is then the problem of finding – if any – $A \in \mathrm{GL}_n(\mathbb{K})$ that makes $\mathbf{g}$ equivalent to $\mathbf{f}$ (*i.e.* $A \in \mathrm{GL}_n(\mathbb{K})$ such that $\mathbf{g}(\mathbf{x}) = \mathbf{f}(A \cdot \mathbf{x})\big)$.

In such a case, we present a randomized polynomial-time algorithm for solving IP1S with quadratic polynomials. To do so, we show that such a problem can be reduced to the variant of a

classical problem of representation theory over finite dimensional algebras. In our setting we need, as in the case $m = 1$, to provide a canonical form of the problem.

Canonical Form of `IP1S`

Let $\big(\mathbf{f} = (f_1, \ldots, f_m), \mathbf{g} = (g_1, \ldots, g_m)\big) \in \mathbb{K}[x_1, \ldots, x_n]^m \times \mathbb{K}[x_1, \ldots, x_n]^m$ be homogeneous quadratic polynomials. Let $H_1, \ldots, H_m$ be the Hessian matrices of $f_1, \ldots, f_m$ (resp. $H'_1, \ldots, H'_m$ be the Hessian matrices of $g_1, \ldots, g_m$). Recall that the Hessian matrix associated to a f_i is defined as $H_i = \left(\frac{\partial^2 f_i}{\partial x_k \partial x_\ell}\right)_{k,\ell} \in \mathbb{K}^{n \times n}$. Consequently, `IP1S` for quadratic forms is equivalent to finding $A \in \mathrm{GL}_n(\mathbb{K})$ such that:

$$H'_i = A^{\mathrm{T}} \cdot H_i \cdot A, \text{ for all } i, 1 \leq i \leq m. \tag{1}$$

Assuming H_j is invertible, and thus so is H'_j, one has $H'^{-1}_j = A^{-1} H_j^{-1} A^{-\mathrm{T}}$. Combining this with equation (1) yields $H'^{-1}_j H'_i = A^{-1} \cdot H_j^{-1} H_i \cdot A$. If none of the H_i's is invertible, then we look for an invertible linear combination thereof. For this reason, we assume all along this paper:

Assumption 1 (Regularity assumption)**.** Let $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{K}[x_1, \ldots, x_n]$. We assume that a linear combination over $\mathbb{K}$ of the quadratic forms $f_1, \ldots, f_m$ is not degenerate[1]. In particular, we assume that $|\mathbb{K}| > n$.

Taking as variables the entries of A, we can see that (1) naturally yields a nonlinear system of equations. However, we show that one can essentially linearize equations (1). To this end, we prove in Section 2 that under Assumption 1 any quadratic homogeneous instance `IP1S` can be reduced, under a randomized process, to a canonical form on which – in particular – all the quadratic forms are nondegenerate. We shall call these instances *regular*. More precisely:

Theorem 1. *Let $\mathbb{K}$ be a field of* $\mathrm{char}\,\mathbb{K} \neq 2$*. There exists a randomized polynomial-time algorithm which given a regular quadratic homogeneous instance of* `IP1S` *returns "*NoSolution*" only if the two systems are not equivalent or a canonical form*

$$\left(\Big(\sum_{i=1}^{n} d_i x_i^2, f_2, \ldots, f_m \Big), \Big(\sum_{i=1}^{n} d_i x_i^2, g_2, \ldots, g_m \Big) \right),$$

where the d_i are equal to 1 *or a nonsquare in $\mathbb{K}$, f_i and g_i are* nondegenerate *homogeneous quadratic polynomials in $\mathbb{K}[x_1, \ldots, x_n]$. Any solution on $\mathbb{K}$ on the canonical form can be efficiently mapped to a solution of the initial instance (and conversely).*

Let us note that over the rationals, computing the exact same sum of squares for the first quadratic forms of each set is difficult, see (Saxena, 2006, Chapter 3), (Wallenborn, 2013, Chapter 1). As such, one could only assume that the first quadratic form of the second set is $\sum_{i=1}^{n} d'_i x_i^2$.

[1] We would like to thank G. Ivanyos for having pointed us this issue in a preliminary version of this paper.

This does not fundamentally change the algorithms presented in this paper, beside some matrices denoted by D which could be changed into $D' = \mathrm{Diag}(d'_1, \ldots, d'_n)$.

Note that the success probability of the algorithms presented here will depend on the size of the field. If one looks for $A \in \mathbb{L}^{n\times n}$ with $\mathbb{L}$ an extension of $\mathbb{K}$, one can amplify the success probability over a small field by using the fact that matrices are conjugate over $\mathbb{K}$ if and only if they are conjugate over an algebraic extension $\mathbb{L}$ (see de Seguins Pazzis (2010)). Thus, one can search linear change of variables with coefficients in some algebraic extension $\mathbb{L} \supseteq \mathbb{K}$ (but of limited degree).

Conjugacy Problem

When `IP1S` is given in canonical form, equations (1) can be rewritten as $A^{\mathrm{T}} DA = D$ with $D = \mathrm{Diag}(d_1, \ldots, d_n)$ and $H'_i = A^{\mathrm{T}} \cdot H_i \cdot A = DA^{-1}D^{-1} \cdot H_i \cdot A$ for all $i, 2 \leq i \leq m$. Our task is now to solve the following problem:

Definition 2 (*D*-**Orthogonal Simultaneous Matrix Conjugacy (***D*-`OSMC`**)).** Let $\mathbb{K}^{n\times n}$ be the set of $n \times n$ matrices with entries in $\mathbb{K}$. Let $\{H_1, \ldots, H_m\}$ and $\{H'_1, \ldots, H'_m\}$ be two families of matrices in $\mathbb{K}^{n\times n}$. The *D*-`OSMC` problem is the task to recover – if any – a D-orthogonal matrix $X \in \mathbb{L}^{n\times n}$, *i.e.* $X^{\mathrm{T}} DX = D$, with $\mathbb{L}$ being an algebraic extension of $\mathbb{K}$, such that:

$$X^{-1} H_i X = H'_i, \qquad \forall i, 1 \leq i \leq m,$$

Chistov et al. (1997) show that *D*-`OSMC` with $D = \mathrm{Id}$ is equivalent to:

1. Solving the Simultaneous Matrix Conjugacy problem (`SMC`) between $\{H_i\}_{1\leq i\leq m}$ and $\{H'_i\}_{1\leq i\leq m}$, that is to say finding an invertible matrix $Y \in \mathrm{GL}_n(\mathbb{K})$ such that:

$$Y^{-1} \cdot H_i \cdot Y = H'_i \quad \text{and} \quad Y^{-1} \cdot H_i^{\mathrm{T}} \cdot Y = H_i'^{\mathrm{T}} \quad \forall i, 1 \leq i \leq m. \tag{2}$$

2. Computing the square-root W of the matrix $Z = Y \cdot Y^{\mathrm{T}}$. Then, the solution of the *D*-`OSMC` problem is given by $X = YW^{-1}$.

In our context, $D = \mathrm{Diag}(d_1, \ldots, d_n)$ is any diagonal invertible matrix. So, we extend Chistov et al. (1997) and show that *D*-`OSMC` is equivalent to

1. Finding an invertible matrix $Y \in \mathrm{GL}_n(\mathbb{K})$ such that:

$$Y^{-1} \cdot H_i \cdot Y = H'_i \quad \text{and} \quad DY^{-1}D^{-1} \cdot H_i^{\mathrm{T}} \cdot DYD^{-1} = H_i'^{\mathrm{T}} \quad \forall i, 1 \leq i \leq m. \tag{3}$$

2. Computing the square-root W of the matrix $Z = DY \cdot Y^{\mathrm{T}} D^{-1}$. Then, the solution of the *D*-`OSMC` problem is given by $X = YW^{-1}$.

In our case, the H_i's (resp. H'_i's) are symmetric (Hessian matrices). Thus, condition (3) yields a system of *linear* equations and one polynomial inequation:

$$H_1 \cdot Y = Y \cdot H'_1, \ldots, H_m \cdot Y = Y \cdot H'_m \text{ and } \det(Y) \neq 0. \tag{4}$$

From now on, we shall denote by $\mathscr{O}_n(\mathbb{L}, D)$ the set of D-orthogonal matrices with coefficients in $\mathbb{L}$.

Let $V \subset \mathbb{K}^{n\times n}$ be the linear subspace of matrices defined by these linear equations. The SMC problem is then equivalent to recovering an invertible matrix in V; in other words we have to solve a particular instance of Edmonds' problem (Edmonds (1967)). Note that, if the representation of the algebra spanned by $\{H_1^{-1}H_i\}_{1\leq i\leq m}$ is *irreducible*, we know that V has dimension at most 1 (Schur's Lemma, see (Lang, 2002, Chap. XVII, Proposition 1.1) and (Newman, 1967, Lemma 2) for a matrix version of this lemma). After putting the equations in triangular form, randomly sampling over the free variables an element in V yields, thanks to Schwartz-Zippel-DeMillo-Lipton Lemma (DeMillo and Lipton (1978); Zippel (1979)), a solution to D-OSMC with overwhelming probability as soon as $\mathbb{K}$ is big enough. If one accepts to have a solution matrix over an extension field, we can amplify the probability of success by considering a bigger algebraic extension (see de Seguins Pazzis (2010)). Whilst a rather "easy" randomized polynomial-time algorithm solves SMC, the task of finding a deterministic algorithm is more delicate. In our particular case, we can adapt the result of Chistov et al. (1997) and provide a deterministic polynomial-time algorithm for solving (2).

Characteristic 2

Let us recall that in characteristic 2, the associated matrices $H_1,\ldots,H_m,H'_1,\ldots,H'_m$ to quadratic forms can be chosen as upper triangular. In this context, we show in Section 3.4 that IP1S can still be reduced to a $(H_1+H_1^{\mathrm{T}})$-conjugacy problem. Under certain conditions in even dimension, we can solve this conjugacy problem in polynomial-time. These results are well confirmed by some experimental results presented in Section 3.5. We can recover a solution in less than one second for n up to one hundred (cryptographic applications of IP1S usually require smaller values of n, typically ≤ 30, for efficiency reasons).

Matrix Square Root Computation

It is well known that computing square roots of matrices can be done efficiently using numerical methods (for instance, see Gantmacher (1959)). On the other hand, it seems difficult to control the bit complexity of numerical methods. In (Chistov et al., 1997, Section 3), the authors consider the problem of computing, in an exact way, the square root of matrices over algebraic number fields. As presented, it is not completely clear that the method proposed is polynomial-time as some coefficients of the result matrix lie in extensions of nonpolynomial size, see Cai (1994). However, by applying a small trick to the proof of Chistov et al. (1997), one can compute a solution in polynomial-time for various field of characteristic $\neq 2$. In any case, for the sake of completeness, we propose two polynomial-time algorithms for this task. First, a general method fixing the issue encountered in (Chistov et al., 1997, Section 3) is presented in Section 3.2. To do so, we adapt the technique of Cai (1994) and compute the square root as the product of two matrices in an algebraic extension which can both be computed in polynomial time. The delicate task being to control the size of the algebraic extensions occurring during the algorithm. In here, each coefficient of the two matrices are lying in an extension field of polynomial degree in n. Furthermore, these matrices allow us to test in polynomial time if $H_1,\ldots,H_m$ and $H'_1,\ldots,H'_m$ are indeed equivalent. We then

present a second simpler method based on the *generalized Jordan normal form* (see Section 6.3) which works (in polynomial time) over finite fields. In general, it deals with algebraic extensions of smaller degree than the first one. Putting things together, we obtain our main result:

Theorem 2. *Let $\mathbb{K}$ be a field with* $\operatorname{char}\mathbb{K} \neq 2$*. Under Assumption 1, there exists a randomized polynomial-time algorithm for solving quadratic-IP1S over an extension field of $\mathbb{K}$ of polynomial degree in n.*

Let us note that the authentication scheme using IP1S requires to find a solution over the base field. However, it is not always necessary to find a solution in the base field (typically, in the context of a key-recovery for multivariate schemes). In Bettale et al. (2013), the authors recover an equivalent key over an extension for the multi-HFE scheme.

In addition, under some nondegeneracy assumption, Theorem 2 can be turned into a deterministic algorithm solving IP1S over the base field $\mathbb{K}$ or an extension thereof. That is:

Theorem 3. *Under Assumption 1 and the assumption that one of the quadratic form is nondegenerate, there is a deterministic polynomial-time algorithm for solving quadratic-IP1S over an extension of $\mathbb{K}$ of polynomial degree in n. Furthermore, if the space of matrices satisfying equations* (4) *has dimension* 1*, then the algorithm can solve quadratic-IP1S over $\mathbb{K}$.*

Let us note that assuming that one of the Hessian matrix is invertible is not a strong assumption when the size of $\mathbb{K}$ is not too small. Indeed, the probability of picking a random invertible symmetric matrix over $\mathbb{F}_q$ is

$$\frac{\prod_{i=1}^{n}(1-q^{-i})}{\prod_{i=1}^{\lfloor n/2\rfloor}(1-q^{-2i})} = \prod_{i=1}^{\lceil n/2\rceil}(1-q^{-2i+1}),$$

see (Carlitz, 1954, Equations 4.7 and 4.8).

If $m \geq 3$, for random matrices $H_1,\ldots,H_m$, the set of solutions of equations (4) is a 1-dimensional matrix space. This allows us to solve quadratic-IP1S in polynomial-time over $\mathbb{K}$. In Section 3.5, we present our timings for solving IP1S. These experiments confirm that for randomly chosen matrices, our method solves IP1S over $\mathbb{K}$. We remark also that our method succeeds to solve IP1S over $\mathbb{F}_2$ for public-keys whose sizes are much bigger than practical ones.

In Section 4, we consider the counting problem #IP1S associated to IP1S for quadratic (homogeneous) polynomials in its canonical form (as defined in Theorem 1). Note that such a counting problem is also related to cryptographic concerns. It corresponds to evaluating the number of equivalent secret-keys in MPKC, see Faugère et al. (2012); Wolf and Preneel (2011). Given homogeneous quadratic polynomials $\big(\mathbf{f}=(f_1,\ldots,f_m),\mathbf{g}=(g_1,\ldots,g_m)\big) \in \mathbb{K}[x_1,\ldots,x_n]^m \times \mathbb{K}[x_1,\ldots,x_n]^m$, we want to count the number of invertible matrices $A \in \mathrm{GL}_n(\mathbb{K})$ such that $\mathbf{g}(\mathbf{x}) = \mathbf{f}(A\cdot\mathbf{x})$. To do so, we define:

Definition 3. Let $\mathbf{f}=(f_1,\ldots,f_m) \in \mathbb{K}[x_1,\ldots,x_n]^m$, we shall call *automorphism group of* $\mathbf{f}$ the set:

$$\mathscr{G}_{\mathbf{f}} = \{A \in \mathrm{GL}_n(\mathbb{K}) \mid \mathbf{f}(A\cdot\mathbf{x}) = \mathbf{f}(\mathbf{x})\}.$$

If $\mathbf{f} \sim \mathbf{g}$, the automorphism groups of $\mathbf{f}$ and $\mathbf{g}$ are similar. Thus, the size of the automorphism group of $\mathbf{f}$ allows us to count the number of invertible matrices mapping $\mathbf{f}$ to $\mathbf{g}$. For quadratic homogeneous polynomials, the automorphism group coincides with the subset of regular matrices in the centralizer $\mathscr{C}(\mathscr{H})$ of the Hessian matrices $\mathscr{H}$ associated to $\mathbf{f}$. Taking α an algebraic element of degree m over $\mathbb{K} = \mathbb{F}_q$, let us assume the Jordan normal form of $H = \sum_{i=i}^{m} \alpha^{i-1} H_i$ has Jordan blocks of sizes $s_{i,1} \leq \cdots \leq s_{i,d_i}$ associated to eigenvalue ζ_i, for $i, 1 \leq i \leq r$. Then, as a consequence of (Singla, 2010, Lemma 4.11), we prove that, if q is an odd prime power, then the number of solutions of quadratic-IP1S in $\mathbb{F}_q^{n \times n}$ is bounded from above by:

$$q^{\left(\sum_{1 \leq i \leq r} \sum_{1 \leq j \leq d_i} (2d_i - 2j+1) s_{i,j}\right)} - 1.$$

Open Question: The Irregular Case

Given a quadratic instance of IP1S, a nondegenerate instance is an instance wherein the matrix whose rows are all the rows of $H_1, \ldots, H_m$ has rank n. In paragraph 2.ii, we see how to transform some degenerate instances into nondegenerate instances. However, nondegenerate instances are not always regular instances. There are cases, the so-called *irregular* cases, such that the vector space of matrices spanned by $H_1, \ldots, H_m$ does not contain a nondegenerate matrix. This situation is well illustrated by the following example $f_1 = x_1 x_3, f_2 = x_2 x_3$. Any linear combination of f_1, f_2 is degenerate, while $\mathbf{f} = (f_1, f_2)$ is not. Note that we can decide in randomized polynomial time if an instance of quadratic-IP1S is irregular since it is equivalent to checking if a determinant is identically equal to zero; thus it is a particular instance of polynomial identity testing. In the irregular case, it is clear that our algorithm fails. In fact, it seems that most known algorithms dedicated to quadratic-IP1S (Bouillaguet et al. (2011); Macario-Rat et al. (2013)) will fail on these instances; making the hardness of the irregular case intriguing and then an interesting open question.

Special case of IP

In our quest of finding instances of IP solvable in polynomial-time, we take a first step in Section 5. We consider IP for a specific set of polynomials with $m = n$. In the aforementioned Section 5, we prove the following:

Theorem 4. *Let $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{K}[x_1, \ldots, x_n]^n$ be given in dense representation, and $\mathbf{f} = \mathbf{POW}_{n,d} = (x_1^d, \ldots, x_n^d) \in \mathbb{K}[x_1, \ldots, x_n]^n$ for some $d > 0$. Whenever $\operatorname{char} \mathbb{K} = 0$, let $e = d$ and $\tilde{\mathbf{g}} = \mathbf{g}$. Otherwise, let $p = \operatorname{char} \mathbb{K}$, let e and r be integers such that $d = p^r e$, p and e coprime, and let $\tilde{\mathbf{g}} \in \mathbb{K}[\mathbf{x}]^n$ be such that $\mathbf{g}(\mathbf{x}) = \tilde{\mathbf{g}}(\mathbf{x}^{p^r})$. Let L be a polynomial size of an arithmetic circuit to evaluate the determinant of the Jacobian matrix of $\tilde{\mathbf{g}}$. If the size of $\mathbb{K}$ is at least $12 \max(2^{L+2}, e(n-1)2^{e(n-1)} + e^3 (n-1)^3, 2(e(n-1)+1)^4)$, then there is a randomized polynomial-time algorithm which recovers – if any – $(A, B) \in \mathrm{GL}_n(\mathbb{K}) \times \mathrm{GL}_n(\mathbb{K})$ such that:*

$$\mathbf{g} = B \cdot \mathbf{POW}_{n,d}(A \cdot \mathbf{x}).$$

This extends a similar result of (Kayal, 2011, Section 5) who considered PolyEquiv for a sum of d-power polynomials. We show that solving IP for $\mathbf{POW}_{n,d}$ reduces to factoring the determinant of a Jacobian matrix (in Kayal (2011), the Hessian matrix is considered). This illustrates, how powerful partial derivatives can be in equivalence problems (Chen et al. (2011); Perret (2005)). To go along with the proof of Theorem 4, we design Algorithm 4 at the end of Section 5.

2. Normalization - Canonical form of IP1S

In this section, we prove Theorem 1. In other words, we explain how to reduce, under Assumption 1, any quadratic homogeneous instance $(\mathbf{f},\mathbf{g}) \in \mathbb{K}[x_1,\ldots,x_n]^m \times \mathbb{K}[x_1,\ldots,x_n]^m$ of IP1S to a suitable canonical form, *i.e.* an instance of IP1S where all the Hessian matrices are invertible and the first two equal the same diagonal invertible matrix. We emphasize that the reduction presented is randomized.

2.i. Homogenization. We show here that the equivalence problem over inhomogeneous polynomials with affine transformation on the variables reduces to the equivalence problem over homogeneous polynomials with linear transformation on the variables. To do so, we simply homogenize the polynomials. Let x_0 be a new variable. For any polynomial $p \in \mathbb{K}[\mathbf{x}]$ of degree 2, we denote by $p^\star(x_0,x_1,\ldots,x_n) = x_0^2\, p(x_1/x_0,\ldots,x_n/x_0)$ its *homogenization*.

Proposition 5. *IP1S with quadratic polynomials and affine transformation on the variables can be reduced in polynomial-time to IP1S with homogeneous quadratic polynomials and linear transformation on the variables.*

Proof. Let $(\mathbf{f},\mathbf{g}) \in \mathbb{K}[\mathbf{x}]^m \times \mathbb{K}[\mathbf{x}]^m$ be inhomogeneous polynomials of degree 2. We consider the transformation which maps $(\mathbf{f},\mathbf{g})$ to $\big(\mathbf{f}^\star = (f_0^\star = x_0^2, f_1^\star,\ldots,f_m^\star), \mathbf{g}^\star = (g_0^\star = x_0^2, g_1^\star,\ldots,g_m^\star)\big)$. This clearly transforms polynomials of degree 2 to homogeneous quadratic polynomials. We can write $f_i(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} H_i\, \mathbf{x} + L_i\, \mathbf{x} + c_i$ with $H_i \in \mathbb{K}^{n\times n}$, $L_i \in \mathbb{K}^n$ and $c_i \in \mathbb{K}$, then $f_i(A\mathbf{x}+b) = (A\mathbf{x}+b)^{\mathrm{T}} H_i\, (A\mathbf{x}+b) + L_i\, (A\mathbf{x}+b) + c_i$ and its homogenization is $(A\mathbf{x}+bx_0)^{\mathrm{T}} H_i\, (A\mathbf{x}+bx_0) + L_i\, (A\mathbf{x}+bx_0)\, x_0 + c_i x_0^2 = f_i^\star(A'\mathbf{x}^\star)$, with $\mathbf{x}^\star = (x_0,x_1,\ldots,x_n)^{\mathrm{T}}$. If $(A,b) \in \mathrm{GL}_n(\mathbb{K}) \times \mathbb{K}^n$ is an affine transformation solution on the inhomogeneous instance then $A' = \left(\begin{smallmatrix} 1 & \mathbf{0} \\ b & A \end{smallmatrix}\right)$ is a solution for the homogenized instance. Conversely, a solution $A' \in \mathrm{GL}_{n+1}(\mathbb{K})$ of the homogeneous problem must stabilize the homogenization variable x_0 in order to be a solution of the inhomogeneous problem. This is forced by adding $f_0 = x_0^2$ and $g_0 = x_0^2$ and setting $C' = A'/a'_{0,0}$, with $a'_{0,0} = \pm 1$. One can see that C' is of the form $\left(\begin{smallmatrix} 1 & \mathbf{0} \\ d & C \end{smallmatrix}\right)$, and $(C,d) \in \mathrm{GL}_n(\mathbb{K}) \times \mathbb{K}^n$ is a solution for $(\mathbf{f},\mathbf{g})$. □

2.ii. Redundant Variables. As a first preliminary natural manipulation, we first want to eliminate – if any – *redundant variables* from the instances considered. Thanks to Carlini (2005) (and reformulated in Kayal (2011)), this task can be done in randomized polynomial time:

Proposition 6. *(Carlini (2005); Kayal (2011)) Let $f \in \mathbb{K}[x_1\ldots,x_n]$ be a polynomial. We shall say that f has s* essential variables *if $\exists M \in \mathrm{GL}_n(\mathbb{K})$ such that $f(M\mathbf{x})$ depends only on the first s variables $x_1,\ldots,x_s$. The remaining $n-s$ variables $x_{s+1}\ldots,x_n$ will be called* redundant *variables. If*

$\mathrm{char}\,\mathbb{K} = 0$ *or* $\mathrm{char}\,\mathbb{K} > \deg f$*, and* f *has* s *essential variables, then we can compute in randomized polynomial time* $M \in \mathrm{GL}_n(\mathbb{K})$ *such that* $f(M\mathbf{x})$ *depends only on the first* s *variables.*

For a quadratic form, s is simply the rank of the associated Hessian matrix. As such, for $m = 1$, a quadratic instance is regular if and only if the associated Hessian matrix is invertible. For a set of equations, we extend the notion of essential variables as follows.

Definition 4. The number of *essential variables* of $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{K}[x_1, \ldots, x_n]^m$ is the smallest s such that $\mathbf{f}$ can be decomposed as:

$$\mathbf{f} = \tilde{\mathbf{f}}(\ell_1, \ldots, \ell_s)$$

with $\ell_1, \ldots, \ell_s$ being linear forms in $x_1, \ldots, x_n$ of rank s and $\tilde{\mathbf{f}} \in \mathbb{K}[y_1, \ldots, y_s]^m$.

The linear forms $\ell_1, \ldots, \ell_s$ can be easily computed thanks to Proposition 6 when the characteristic of $\mathbb{K}$ is zero or greater than the degrees of $f_1, \ldots, f_m$. In characteristic 2, when $\mathbb{K}$ is perfect (which is always true if $\mathbb{K}$ is finite for instance) the linear forms can also be recovered in polynomial time (see Berthomieu et al. (2010); Giraud (1972); Hironaka (1970) for instance). Below, we show that we can restrict our attention to only essential variables. Namely, solving `IP1S` on $(\mathbf{f}, \mathbf{g})$ reduces to solving `IP1S` on instances having only essential variables.

Proposition 7. *Let* $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[x_1, \ldots, x_n]^m \times \mathbb{K}[x_1, \ldots, x_n]^m$ *be two sets of quadratic polynomials. If* $\mathbf{f} \sim \mathbf{g}$*, then their numbers of essential variables must be the same. Let* s *be the number of essential variables of* $\mathbf{f}$*. Finally, let* $(\tilde{\mathbf{f}}, \tilde{\mathbf{g}}) \in \mathbb{K}[y_1, \ldots, y_s]^m \times \mathbb{K}[y_1, \ldots, y_s]^m$ *be such that:*

$$\mathbf{f} = \tilde{\mathbf{f}}(\ell_1, \ldots, \ell_s) \text{ and } \mathbf{g} = \tilde{\mathbf{g}}(\ell'_1, \ldots, \ell'_s),$$

with $\ell_1, \ldots, \ell_s$ *(resp.* $\ell'_1, \ldots, \ell'_s$*) linear forms in* $\mathbf{x}$ *of rank* s *and* $\tilde{\mathbf{f}}, \tilde{\mathbf{g}} \in \mathbb{K}[y_1, \ldots, y_s]^m$*. It holds that:*

$$\mathbf{f} \sim \mathbf{g} \iff \tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}.$$

Proof. Let $H_1, \ldots, H_m$ be the Hessian matrices of $f_1, \ldots, f_m$ (resp. $H'_1, \ldots, H'_m$ be the Hessian matrices of $g_1, \ldots, g_m$). Similarly, we define the Hessian matrices $\tilde{H}_1, \ldots, \tilde{H}_m$ (resp. $\tilde{H}'_1, \ldots, \tilde{H}'_m$) of $\tilde{f}_1, \ldots, \tilde{f}_m$ (resp. $\tilde{g}_1, \ldots, \tilde{g}_m$). Let also M and N be matrices in $\mathrm{GL}_n(\mathbb{K})$ such that $H_i = M^{\mathrm{T}} \begin{pmatrix} \tilde{H}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} M$ and $H'_i = N^{\mathrm{T}} \begin{pmatrix} \tilde{H}'_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} N$ for all $i, 1 \leq i \leq m$. There exist such M and N, as $\mathbf{f}$ and $\mathbf{g}$ have essentially s variables. Up to re-indexing the rows and columns of H_i and H'_i, so that they remain symmetric, one can always choose M and N such that $M = \begin{pmatrix} M_1 & M_2 \\ \mathbf{0} & \mathrm{Id} \end{pmatrix}$ and $N = \begin{pmatrix} N_1 & N_2 \\ \mathbf{0} & \mathrm{Id} \end{pmatrix}$, with $M_1, N_1 \in \mathrm{GL}_s(\mathbb{K})$.

If $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$, $\exists \tilde{A} \in \mathrm{GL}_s(\mathbb{K})$ such that $A^{\mathrm{T}} \tilde{H}_i \tilde{A} = \tilde{H}'_i$, for all $i, 1 \leq i \leq m$. Then, for all $B \in \mathbb{K}^{(n-s)\times s}$ and $C \in \mathrm{GL}_{n-s}(\mathbb{K})$:

$$\begin{pmatrix} \tilde{A}^{\mathrm{T}} & B^{\mathrm{T}} \\ \mathbf{0} & C^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \tilde{H}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \tilde{A} & \mathbf{0} \\ B & C \end{pmatrix} = \begin{pmatrix} \tilde{H}'_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

$$N^{\mathrm{T}} \begin{pmatrix} \tilde{A}^{\mathrm{T}} & B^{\mathrm{T}} \\ \mathbf{0} & C^{\mathrm{T}} \end{pmatrix} M^{-\mathrm{T}} H_i M^{-1} \begin{pmatrix} \tilde{A} & \mathbf{0} \\ B & C \end{pmatrix} N = H'_i.$$

Therefore, $\mathbf{f}$ and $\mathbf{g}$ are equivalent.

Conversely, we assume now that $\mathbf{f} \sim \mathbf{g}$, *i.e.* there exists $A \in \mathrm{GL}_n(\mathbb{K})$ such that $A^{\mathrm{T}} \cdot H_i \cdot A = H'_i$, for all $i, 1 \leq i \leq m$. This implies that:

$$N^{-\mathrm{T}} A^{\mathrm{T}} M^{\mathrm{T}} \begin{pmatrix} \tilde{H}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} MAN^{-1} = \begin{pmatrix} \tilde{H}'_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \forall i, 1 \leq i \leq m.$$

We then define $\tilde{A} = ((MAN^{-1})_{i,j})_{1 \leq i,j \leq s}$, so that $\tilde{\mathbf{f}}(\tilde{A}\mathbf{x}) = \tilde{\mathbf{g}}(\mathbf{x})$. As $\mathbf{g}$ has s essential variables, then $\operatorname{rank} \tilde{A}$ cannot be smaller than s, hence $\tilde{A} \in \mathrm{GL}_s(\mathbb{K})$. We then get $\tilde{A}^{\mathrm{T}} \tilde{H}_i \tilde{A} = \tilde{H}'_i$ for all $i, 1 \leq i \leq m$, *i.e.* $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$. □

According to Proposition 7, there is an efficient reduction mapping an instance $(\mathbf{f}, \mathbf{g})$ of `IP1S` to an instance $(\tilde{\mathbf{f}}, \tilde{\mathbf{g}})$ of `IP1S` having only essential variables. From now on, we will then assume that we consider instances of `IP1S` with n essential variables for both $\mathbf{f}$ and $\mathbf{g}$.

2.iii. Canonical Form. We now assume that $\operatorname{char} \mathbb{K} \neq 2$.

Definition 5. Let $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{K}[x_1, \ldots, x_n]^m$ be quadratic homogeneous forms with Hessian matrices $H_1, \ldots, H_m$. We shall say that $\mathbf{f}$ is *regular* if its number of essential variables is n and if $\exists \lambda_1, \ldots, \lambda_m \in \mathbb{K}$ such that $\det(\sum_{i=1}^m \lambda_i H_i) \neq 0$.

Remark 8. Our algorithm requires that amongst all the Hessian matrices, one at least is invertible, the so-called regular case. It is not sufficient to only assume that the number of essential variables is n. Indeed, Ivanyos's irregular example $\mathbf{f} = (x_1 x_3, x_2 x_3)$ has 3 essential variables, but any nonzero linear combination $\lambda_1 f_1 + \lambda_2 f_2$ has only 2 essential variables $\lambda_1 x_1 + \lambda_2 x_2$ and x_3. Similarly, $\mathbf{f} = (x_1^2 + x_2^2 + x_3^2, x_2^2 + 2x_3^2 + x_4^2)$ has 4 essential variables but any nonzero linear combination $\lambda_1 f_1 + \lambda_2 f_2$ over $\mathbb{F}_3$ has only 3 essential variables. This explains the additional condition on the previous definition, and our Assumption 1.

We are now in a position to reduce quadratic homogeneous instances of `IP1S` to a first simplified form.

Proposition 9. *Let $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[x_1, \ldots, x_n]^m \times \mathbb{K}[x_1, \ldots, x_n]^m$ be regular quadratic homogeneous polynomials. There is a randomized polynomial-time algorithm which returns "*NoSolution*" only if $\mathbf{f} \nsim \mathbf{g}$, or a new instance*

$$(\tilde{\mathbf{f}}, \tilde{\mathbf{g}}) = \left(\Big(\sum_{i=1}^n d_i x_i^2, \tilde{f}_2, \ldots, \tilde{f}_m \Big), \Big(\sum_{i=1}^n d_i x_i^2, \tilde{g}_2, \ldots, \tilde{g}_m \Big) \right) \in \mathbb{K}[\mathbf{x}]^m \times \mathbb{K}[\mathbf{x}]^m,$$

with $d_1, \ldots, d_n$ being 1 or nonsquares in $\mathbb{K}$, such that $\mathbf{f} \sim \mathbf{g} \iff \tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$. If $\mathbb{K}$ is finite, the output of this algorithm is correct with probability at least $1 - n/|\mathbb{K}|$. If $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$, invertible matrices P, Q and $A' \in \mathrm{GL}_n(\mathbb{K})$ are returned such that $\mathbf{f}(P\mathbf{x}) = \tilde{\mathbf{f}}(\mathbf{x})$, $\mathbf{g}(Q\mathbf{x}) = \tilde{\mathbf{g}}(\mathbf{x})$ and $\tilde{\mathbf{f}}(A'\mathbf{x}) = \tilde{\mathbf{g}}(\mathbf{x})$. It then holds that $\mathbf{f}(PA'Q^{-1}\mathbf{x}) = \mathbf{g}(\mathbf{x})$.

Proof. Let $H_1,\ldots,H_m$ be the Hessian matrices associated to $f_1,\ldots,f_m$. According to Schwartz-Zippel-DeMillo-Lipton Lemma (DeMillo and Lipton (1978); Zippel (1979)), we can compute in randomized polynomial time $\lambda_1,\ldots,\lambda_m \in \mathbb{K}$ such that $\varphi = \sum_{i=1}^m \lambda_i \cdot f_i$ is regular, *i.e.* $\det(\sum_{i=1}^m \lambda_i H_i) \neq 0$. The probability to pick $(\lambda_1,\ldots,\lambda_m) \in \mathbb{K}^m$ on which φ is regular is bounded from above by $n/|\mathbb{K}|$. We define $\gamma = \sum_{i=1}^m \lambda_i \cdot g_i$. Should one reorder the equations, we can assume w.lo.g. that $\lambda_1 \neq 0$. We have then:

$$\mathbf{f} \sim \mathbf{g} \iff (\varphi, f_2, \ldots, f_m) \sim (\gamma, g_2, \ldots, g_m).$$

Now, applying Gauß's reduction algorithm to φ, there exists $d_1,\ldots,d_n \in \mathbb{K}$, each being 1 or a nonsquare, such that $\varphi = \sum_{i=1}^n d_i \ell_i^2$, where $\ell_1,\ldots,\ell_n$ are independent linear forms in $x_1,\ldots,x_n$. This gives a $P \in \mathrm{GL}_n(\mathbb{L})$ such that $\tilde{\mathbf{f}} = (\tilde{\varphi} = \sum_{i=1}^n d_i x_i^2, \tilde{f_2}, \ldots, \tilde{f_m}) = (\varphi(P\mathbf{x}), f_2(P\mathbf{x}), \ldots, f_m(P\mathbf{x}))$. Clearly, $\mathbf{f} \sim \tilde{\mathbf{f}}$, hence, $\tilde{\mathbf{f}} \sim \mathbf{g}$.

After that, we can apply once again Gauß's reduction algorithm to γ. If the reduced polynomial is different from $\sum_{i=1}^n d_i x_i^2$, then $\mathbf{f} \nsim \mathbf{g}$ and we return "NoSolution". Otherwise, the reduction is given by a matrix $Q \in \mathrm{GL}_n(\mathbb{L})$ such that $\tilde{\mathbf{g}} = (\tilde{\gamma} = \sum_{i=1}^n d_i x_i^2, \tilde{g}_2, \ldots \tilde{g}_m) = (\gamma(Q\mathbf{x}), g_2(Q\mathbf{x}), \ldots, g_m(Q\mathbf{x}))$ and $\mathbf{g} \sim \tilde{\mathbf{g}}$. Thus, $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$ if and only if $\mathbf{f} \sim \mathbf{g}$.

Now, assume that $\exists A' \in \mathrm{GL}_n(\mathbb{K})$ such that $\tilde{\mathbf{f}}(A'\mathbf{x}) = \tilde{\mathbf{g}}(\mathbf{x})$. Then, $\mathbf{f}(PA'\mathbf{x}) = \mathbf{g}(Q\mathbf{x})$, *i.e.* $\mathbf{f}(PA'Q^{-1}\mathbf{x}) = \mathbf{g}(\mathbf{x})$. □

Let us recall that whenever $\mathbb{K} = \mathbb{Q}$, computing the exact same sum of squares for $\tilde{f_1}$ and $\tilde{g_1}$ is difficult, see (Saxena, 2006, Chapter 3), (Wallenborn, 2013, Chapter 1). As such, we could only assume that our canonical form is $\tilde{g_1} = \sum_{i=1}^n d_i' x_i^2$. This would merely change the formulation of following Theorem 10.

2.iv. Invertible Hessian Matrices. We are now in a position to reduce any regular homogeneous quadratic instances $(\mathbf{f},\mathbf{g})$ of `IP1S` to a new form of the instances where all the polynomials are themselves regular assuming we could find one. From Proposition 9, this is already the case – under randomized reduction – for f_1 and thus g_1. For the other polynomials, we proceed as follows. For $i, 2 \leq i \leq m$, if the Hessian matrix H_i of f_i is invertible, then we do nothing. Otherwise, we change H_i into $H_i - \nu_i H_1$, with ν_i not an eigenvalue of $H_i H_1^{-1}$. As $\mathbb{K}$ has at least $n+1$ elements, there exists such a ν_i in $\mathbb{K}$. This gives the following result:

Theorem 10. *Let $(\mathbf{f},\mathbf{g}) \in \mathbb{K}[x_1,\ldots,x_n]^m \times \mathbb{K}[x_1,\ldots,x_n]^m$ be regular quadratic homogeneous polynomials. There is a randomized polynomial-time algorithm which returns "*NoSolution*" only if $\mathbf{f} \nsim \mathbf{g}$. Otherwise, the algorithm returns two sets of $n \times n$ invertible symmetric matrices $\{D, \tilde{H}_2 \ldots, \tilde{H}_m\}$ and $\{D, \tilde{H}_2', \ldots, \tilde{H}_m'\}$, with D diagonal, defined over $\mathbb{K}$ such that:*

$$\mathbf{g}(\mathbf{x}) = \mathbf{f}(A\,\mathbf{x}),\ \textit{for } A \in \mathrm{GL}_n(\mathbb{K}) \iff \begin{array}{l} A'^{-1} D^{-1} \tilde{H}_i A' = D^{-1} \tilde{H}_i', \forall i, 1 \leq i \leq m, \\ \textit{for } A' \in \mathscr{O}_n(\mathbb{K}, D), \end{array}$$

with $\mathscr{O}_n(\mathbb{K},D)$ denoting the set of $n \times n$ D-orthogonal matrices over $\mathbb{K}$.

Proof. Combining Proposition 9 and paragraph 2.iv any regular quadratic homogeneous instance of IP1S can be reduced in randomized polynomial time to "NoSolution", only if the two systems are not equivalent, or to a

$$(\tilde{\mathbf{f}},\tilde{\mathbf{g}}) = \left((\sum_{i=1}^{n} d_i x_i^2, \tilde{f}_2, \ldots, \tilde{f}_m), (\sum_{i=1}^{n} d_i x_i^2, \tilde{g}_2, \ldots, \tilde{g}_m) \right),$$

where all the polynomials are *nondegenerate* homogeneous quadratic polynomials in $\mathbb{K}[\mathbf{x}]$. It follows that $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}} \iff \exists A' \in \mathrm{GL}_n(\mathbb{K})$ such that $\forall i, 1 \le i \le m$, ${A'}^{\mathrm{T}} \tilde{H}_i A' = \tilde{H}'_i$. In particular ${A'}^{\mathrm{T}} D A' = D$ and A' is D-orthogonal. Hence, ${A'}^{\mathrm{T}} \tilde{H}_i A' = D A'^{-1} D^{-1} \tilde{H}_i A' = \tilde{H}'_i$, $\forall i, 1 \le i \le m$. □

The proof of this result implies Theorem 1.

2.v. Field Extensions and Jordan Normal Form. To amplify the success probability of our results, it will be convenient to embed a field $\mathbb{F}$ in some finite extension $\mathbb{F}'$ of $\mathbb{F}$. This is motivated by the fact that matrices in $\mathbb{F}^{n\times n}$ are similar if and only if they are similar in $\mathbb{F}'^{n\times n}$, see de Seguins Pazzis (2010). In this paper, we will need to compute the *Jordan normal form* J of some matrix H in several situations. The computation of the Jordan normal form is done in two steps. First, we factor the characteristic polynomial, using for instance Berlekamp's algorithm over $\mathbb{F} = \mathbb{F}_q$ in $O(n\mathsf{M}(n)\log(qn))$ operations in $\mathbb{F}$, where $\mathsf{M}(n)$ is a bound on the number of operations in $\mathbb{F}$ to multiply two polynomials in $\mathbb{F}[x]$ of degree at most $n-1$, see (von zur Gathen and Gerhard, 1999, Theorem 14.14). Then, we use Storjohann (1998)'s algorithm to compute the generalized eigenvectors in $O(n^\omega)$ operations in $\mathbb{F}$, with ω being the exponent of time complexity of matrix multiplication, $2 \le \omega \le 3$.

3. Quadratic IP1S

In this section, we present efficient algorithms for solving regular quadratic-IP1S. According to Proposition 5, we can w.l.o.g. restrict our attention on linear changes of variables and homogeneous quadratic instances. Let D be a diagonal invertible matrix with 1 or nonsquare elements on the diagonal. Let $\mathscr{H} = \{D, H_2, \ldots, H_m\}$ and $\mathscr{H}' = \{D, H'_2, \ldots, H'_m\}$ be two families of invertible symmetric matrices in $\mathbb{K}^{n\times n}$. As explained in Theorem 10, our task reduces – under a randomized process – to finding a D-orthogonal matrix $A' \in \mathscr{O}_n(\mathbb{K}, D)$ such that:

$$A'^{-1} D^{-1} H_i A' = D^{-1} H'_i, \ \forall i, 1 \le i \le m. \tag{5}$$

Case $D = \mathrm{Id}$ was studied in (Chistov et al., 1997, Theorem 4). The authors prove that there is an orthogonal solution A, such that $H_i A = A H'_i$ if and only if there is an invertible matrix Y such that $H_i Y = Y H'_i$ and $H_i^{\mathrm{T}} Y = Y H_i^{\mathrm{T}}$. In our case, whenever $D = \mathrm{Id}$, the matrices are symmetric. So, the added conditions – with the transpose – are automatically fulfilled. In Chistov et al. (1997), the authors suggest then to use the polar decomposition of $Y = AW$, with W symmetric and A orthogonal. Then, A is an orthogonal solution of (5).

The main idea to compute A is to compute W as the square root of $Z = Y^{\mathrm{T}} Y$ as stated in (Chistov et al., 1997, Section 3). However, in general W and A are not defined over $\mathbb{K}$ but over $\mathbb{L} = \mathbb{K}(\sqrt{\zeta_1}, \ldots, \sqrt{\zeta_r})$, where $\zeta_1, \ldots, \zeta_r$ are the eigenvalues of Z. Assuming ζ_1 is the root of an irreducible polynomial P of degree d, then $\zeta_2, \ldots, \zeta_d$ are also roots of the same polynomial. However, there is no reason for them to be in $\mathbb{K}[x]/(P) = \mathbb{K}(\zeta_1)$. But they will be the roots of a polynomial of degree $d-1$, in general, over the field $\mathbb{K}(\zeta_1)$. Then, doing another extension might only add one eigenvalue in the field. Repeating this process yields a field of degree $d!$ over $\mathbb{K}$. As a consequence, in the worst case, we can have to work over an extension field of degree $n!$. Therefore, computing W could be the bottleneck of the method.

Chistov et al. (1997) emphasize that constructing such a square root W in polynomial time is the only serious algorithmic problem. As presented, it is not completely clear that the method proposed is efficient. They propose to compute $W = \sqrt{Y^{\mathrm{T}} Y}$ and then to set $A = W^{-1} Y$. According to Cai's work (Cai (1994)), some coefficients of matrix A may lie in an extension of exponential degree. *Blockwise computation* (see the proof of Proposition 12) can allow us to compute such a matrix. Chistov, Ivanyos and Karpinski set y_i as the restriction of Y to the ith eigenspace, associated to ζ_i, of $Y^{\mathrm{T}} Y$. Then, $x_i = \sqrt{\zeta_i}^{-1} y_i$ and they return the block diagonal matrix constructed from the x_i's. However, this construction gives the impression that the ith eigenspace of $Y^{\mathrm{T}} Y$ is stable by Y, as W^{-1} would act as a multiplication by $\sqrt{\zeta_i}^{-1}$. As a consequence, the blockwise computation was not ensured.

However, this issue does not happen if one uses the same proof on $W = \sqrt{Y Y^{\mathrm{T}}}$ and $A = Y W^{-1}$. In the following subsection, we extend their proof to any invertible diagonal matrix D.

3.1. Existence of a D-Orthogonal Solution

The classical polar decomposition is used in (Chistov et al., 1997, Theorem 4) to determine an orthogonal solution. Using the analogous decomposition, the so-called Generalized Polar Decomposition (GPD), which depends on D, yields a D-orthogonal solution, see Mackey et al. (2005). The GPD of an invertible matrix Y is the factorization $Y = AW$, with A D-orthogonal and W in the associated Jordan algebra, *i.e.* $W^{\mathrm{T}} = DWD^{-1}$. Let us notice that A and W might be defined only over $\mathbb{K}'$ an algebraic extension of $\mathbb{K}$ of some degree.

Proposition 11. *Let $\mathscr{K} = \{K_1, \ldots, K_m\}$ and $\mathscr{K}' = \{K'_1, \ldots, K'_m\}$ be two subsets of m matrices in $\mathbb{K}^{n \times n}$. Let D be an invertible diagonal matrix. There is a D-orthogonal solution $A \in \mathbb{K}'^{n \times n}$ to the conjugacy problem $K_i A = A K'_i$ for all $1 \leq i \leq m$, if and only if there is an invertible solution $Y \in \mathbb{K}'^{n \times n}$ to the conjugacy problem $K_i Y = Y K'_i$ and $K_i^{\mathrm{T}} DYD^{-1} = DYD^{-1} K_i'^{\mathrm{T}}$ for all $1 \leq i \leq m$. Furthermore, if $Y = AW$ is the* GPD *of Y with respect to D, then A suits.*

Proof. This proof is a generalization of (Chistov et al., 1997, Section 3). If A is a D-orthogonal solution to the first problem, then as $A^{\mathrm{T}} = DA^{-1}D^{-1}$, it is clear that A is a solution to the second problem. Conversely, let Y be a solution to the second problem, then $Z = D^{-1} Y^{\mathrm{T}} DY$ commutes with K'_i. As Y is invertible, so is Z, therefore, given a determination of the square roots of the eigenvalues of Z, there is a unique matrix W with these eigenvalues such that $W^2 = Z$ and W is in

the Jordan algebra associated to D, that is $W^{\mathrm{T}} = DWD^{-1}$, see (Mackey et al., 2005, Theorem 6.2). As such, W is a polynomial in Z as proven in Section 6.1 and commutes with K_i'.

Finally, $A = YW^{-1}$ is an D-orthogonal solution of the first problem. As W commutes with K_i', $A^{-1}K_iA = WY^{-1}K_iYW^{-1} = WK_i'W^{-1} = K_i'$ and

$$A^{\mathrm{T}}DA = W^{-\mathrm{T}}Y^{\mathrm{T}}DYW^{-1} = DW^{-1}D^{-1}Y^{\mathrm{T}}DYW^{-1} = DW^{-1}ZW^{-1} = D. \qquad \square$$

For the sake of completeness, we present several efficient algorithms for performing the square root computation.

3.2. Computing the D-Orthogonal Solution

The goal of this part is to "D-orthogonalize" an invertible solution $Y \in \mathrm{GL}_n(\mathbb{K})$ of equation (5). Instead of computing exactly $A \in \mathscr{O}_n(\mathbb{L}, D)$, we compute in polynomial time two matrices whose product is A. These matrices allow us to verify in polynomial time that H_i and H_i' are equivalent for all $i, 1 \leq i \leq m$. To be more precise, we prove the following proposition.

Proposition 12. *Let $\mathscr{H} = \{H_1 = D, H_2, \ldots, H_m\}$ and $\mathscr{H}' = \{H_1' = D, H_2', \ldots, H_m'\}$ be two sets of invertible matrices in $\mathbb{K}^{n\times n}$. We can compute in polynomial time two matrices representations of matrices S and T defined over an algebraic extension $\mathbb{L}$ such that ST^{-1} is D-orthogonal and for all $1 \leq i \leq m$, $D^{-1}H_i(ST^{-1}) = (ST^{-1})D^{-1}H_i'$. In the worst case, product ST^{-1} cannot be computable in polynomial time over $\mathbb{L}$. However, matrices $S^{\mathrm{T}}H_iS$ and $T^{\mathrm{T}}H_i'T$ can be computed and tested for equality in polynomial time.*

Proof. Let $Y \in \mathrm{GL}_n(\mathbb{K})$ such that $D^{-1}H_iY = YD^{-1}H_i'$, $\forall i, 1 \leq i \leq m$. We set $Z = D^{-1}Y^{\mathrm{T}}DY$. Let us denote by T, the change of basis matrix such that $J = T^{-1}ZT$ is the Jordan normal form of Z. According to Cai (1994), T, T^{-1} and J can be computed in polynomial time. Because of the issue of mixing all the eigenvalues of Z, we cannot compute efficiently A in one piece. We will then compute AT and T^{-1} separately. Indeed, AT (resp. T^{-1}) is such that each of its columns (resp. each of its rows) is defined over an extension field $\mathbb{K}(\zeta_i)$, where $\zeta_1, \ldots, \zeta_r$ are the eigenvalues of Z.

We shall say that a matrix is *block-wise* (resp. *columnblock-wise*, *rowblock-wise*) *defined over* $\mathbb{K}(\zeta)$ if for all $1 \leq i \leq r$, its ith block (resp. block of columns, block of rows) is defined over $\mathbb{K}(\zeta_i)$. The size of the ith block being the size of the ith Jordan block of J.

As $J = T^{-1}ZT$ is a Jordan normal form, it is block-wise defined over $\mathbb{K}(\zeta)$. Using the closed formula of Section 6.1, one can compute in polynomial time a square root G of J. This matrix is a block diagonal matrix, block-wise defined over $\mathbb{K}(\sqrt{\zeta})$, hence it can be inverted in polynomial time. Should one want W, one would have to compute $W = TGT^{-1}$. Let us recall that matrices T and T^{-1} are respectively columnblock-wise and rowblock-wise defined over $\mathbb{K}(\zeta)$, see (Cai, 1994, Section 4). Since Y is defined over $\mathbb{K}$, then YT is columnblock-wise defined over $\mathbb{K}(\zeta)$. Thus $S = AT = YW^{-1}T = YTG^{-1}$ is columnblock-wise defined over $\mathbb{K}(\sqrt{\zeta})$. We recall that product $AT \cdot T^{-1}$ mangles the eigenvalues and make each coefficient defined over $\mathbb{K}(\sqrt{\zeta_1}, \ldots, \sqrt{\zeta_r})$ and thus must be avoided.

Now, to verify that $A^{\mathrm{T}}HA = H'$, for any $H \in \mathscr{H}$ and the corresponding $H' \in \mathscr{H}'$, we compute separately $S^{\mathrm{T}}HS = T^{\mathrm{T}}A^{\mathrm{T}}HAT$ and $T^{\mathrm{T}}H'T$. For the former, $S = AT$ (resp. $S^{\mathrm{T}} = (AT)^{\mathrm{T}}$) is

columnblock-wise (resp. rowblock-wise) defined over $\mathbb{K}(\sqrt{\zeta})$ and H is defined over $\mathbb{K}$. Therefore, the product matrix makes each of the coefficients which are on both the ith block of rows and the jth block of columns defined over $\mathbb{K}(\sqrt{\zeta_i}, \sqrt{\zeta_j})$ and so can be computed in polynomial time. For the latter, the same behaviour occurs on the resulting matrix as T is columnblock-wise defined over $\mathbb{K}(\zeta)$. □

Let us assume that the characteristic polynomial of Z, of degree n, can be factored as $P_1^{e_1} \cdots P_s^{e_s}$ with P_i and P_j coprime whenever $i \neq j$, $\deg P_i = d_i$ and $e_i \geq 1$. From a computation point of view, one needs to introduce a variable $\alpha_{i,j}$ for each root of P_i and then a variable $\beta_{i,j}$ for the square root of $\alpha_{i,j}$. This yields a total number of $2\sum_{i=1}^{s} d_i$ variables. In Section 6.3, we present another method which manages to introduce only $2s$ variables in characteristic $p > 2$.

3.3. Probabilistic and Deterministic Algorithms

We first describe a simple probabilistic algorithm summarizing the method of Section 3.2.

Algorithm 1. Probabilistic algorithm.

Input Two sets of invertible symmetric matrices $\mathscr{H} = \{H_1 = D, \ldots, H_m\} \subseteq \mathbb{K}^{n\times n}$ and $\mathscr{H}' = \{H_1' = D, \ldots, H_m'\} \subseteq \mathbb{K}^{n\times n}$.

Output A description of the matrix $A \in \mathrm{GL}_n(\mathbb{L})$ such that $H_i' = A^{\mathrm{T}} H_i A$ for all $1 \leq i \leq m$ or "NoSolution".

1. Compute the vector subspace $\mathscr{Y} = \{Y \mid D^{-1} H_i Y = Y D^{-1} H_i', \forall 1 \leq i \leq m\} \subseteq \mathbb{K}^{n\times n}$.
2. **If** $\mathscr{Y}$ is reduced to the null matrix **then return** "NoSolution".
3. Pick at random $Y \in \mathscr{Y}$.
4. Compute $Z = D^{-1} Y^{\mathrm{T}} D Y$ and $J = T^{-1} Z T \in \mathbb{L}^{n\times n}$, the Jordan normal form of Z together with T.
5. Compute G^{-1} the inverse of a square root of J.
6. **Return** $Y T G^{-1}$ and T.

Theorem 13. *Algorithm 1 is correct with probability at least* $1 - n/|\mathbb{K}|$ *and runs in polynomial time.*

Proof. The correctness and the polynomial-time complexity of the algorithm come from Section 3.2. After computing $\mathscr{Y}$ and putting the equations defining its matrices in triangular form, one has to pick at random one matrix $Y \in \mathscr{Y}$. By sampling the whole field $\mathbb{K}$ on these free variables, the probability that $\det Y = 0$ is upper bounded by $n/|\mathbb{K}|$ thanks to Schwartz-Zippel-DeMillo-Lipton Lemma (DeMillo and Lipton (1978); Zippel (1979)). □

Remark 14. Let us recall that the conjugacy problem does not depend on the ground field (see de Seguins Pazzis (2010)), *i.e.* if there exists $Y \in \mathrm{GL}_n(\mathbb{K}')$, such that $H_i Y = Y H_i'$, then there exists $Y' \in \mathrm{GL}_n(\mathbb{K})$ such

that $H_iY' = Y'H'_i$. This allows us to extend $\mathbb{K}$ to a finite extension in order to decrease the probability of getting a singular matrix Y. Thus the success probability of Algorithm 1 can be amplified to $1 - n/|\mathbb{K}'|$ for any extension $\mathbb{K}' \supseteq \mathbb{K}$. The probability can be then made overwhelming large by considering extension of degree $O(n)$. In this case, the algorithm returns the description of a solution on $\mathbb{K}'(\sqrt{\zeta_1}, \ldots, \sqrt{\zeta_r})$. Notice also that this algorithm can be turned into a deterministic algorithm using (Chistov et al., 1997, Theorem 2). That is, there is a polynomial-time algorithm allowing to compute an invertible element in $\mathscr{Y}$. Furthermore, if one of the original Hessian matrices is already invertible, the computations of the essential variables of paragraph 2.ii and the search of an equation with n essential variables in paragraph 2.iii can be done in a deterministic way. Whence, the whole algorithm is deterministic.

The main Theorem 3 summarizes this remark together with Theorem 13.

3.4. The binary Case

In this section, we investigate fields of characteristic 2. Let $\mathbb{K} = \mathbb{F}_q$ and $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[\mathbf{x}]^m \times \mathbb{K}[\mathbf{x}]^m$. Instead of Hessian matrices, we consider equivalently upper triangular matrices $H_1, \ldots, H_m$ and $H'_1, \ldots, H'_m$ such that:

$$f_i(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} H_i \mathbf{x},\ g_i(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} H'_i \mathbf{x}, \quad \forall\, 1 \leq i \leq m.$$

For any matrix $M \in \mathbb{K}^{n\times n}$, let us denote $\Delta(M) = \mathrm{Diag}(m_{11}, \ldots, m_{nn})$ and $\Sigma(M) = M + M^{\mathrm{T}}$. It is classical that if there exists $A \in \mathrm{GL}_n(\mathbb{K})$ such that $\mathbf{g}(\mathbf{x}) = \mathbf{f}(A \cdot \mathbf{x})$, then we also have

$$\Sigma(H'_i) = A^{\mathrm{T}} \Sigma(H_i) A, \tag{6}$$

$$\Delta(H'_i) = \Delta(A^{\mathrm{T}} H_i A), \quad \forall\, i,\ 1 \leq i \leq m. \tag{7}$$

It suffices for this to expand $\mathbf{f}(A \cdot \mathbf{x})$ and to consider the upper triangular matrices. In a sense, $\Sigma(H_i)$ is the Hessian matrix of f_i and $\Delta(H_i)$ allow us to remember the x_j^2 terms in f_i. Combining two equations of (6) yields $\Sigma(H'_j)^{-1}\Sigma(H'_i) = A^{-1}\Sigma(H_j)^{-1}\Sigma(H_i)A$ as long as $\Sigma(H_j)$ is invertible. Let us notice that $\Sigma(H_i)$'s are symmetric matrices with a zero diagonal, thus antisymmetric matrices with a zero diagonal. We would like to stress out that in odd dimension, the determinant of a symmetric matrix S with a zero diagonal is always zero. Indeed, expanding formula $\sum_{\sigma\in\mathfrak{S}_n} \prod_{i=1}^n s_{i,\sigma(i)}$ yields, for each nonzero term $\prod_{i=1}^n s_{i,\sigma(i)}$, the term $\prod_{i=1}^n s_{i,\sigma^{-1}(i)} = \prod_{i=1}^n s_{\sigma(i),i} = \prod_{i=1}^n s_{i,\sigma(i)}$. Dimension n being odd, $\prod_{i=1}^n s_{i,\sigma(i)}$ cannot be the same term as $\prod_{i=1}^n s_{i,\sigma^{-1}(i)}$. Hence they cancel each other. One can also see these matrices as projections of antisymmetric matrices over a ring of characteristic 0, namely $\mathbb{Z}_q$ the unramified extension of the ring of dyadic integers of degree $\log_2 q$. Let $\tilde{S} \in \mathbb{Z}_q^{n\times n}$ be antisymmetric such that $\tilde{S} \mapsto S$. Then $\det\tilde{S} = \det\tilde{S}^{\mathrm{T}} = \det(-\tilde{S}) = (-1)^n \det\tilde{S}$, hence $\det\tilde{S} = 0$ and $\det S = 0$.

Therefore, if n is odd, then a linear combination of the $\Sigma(H_i)$'s will always be singular. This can be related to the *irregular case* of the introduction.

Reduction to canonical representations in even dimension. In this setting, we also rely on Assumption 1 to assume that a linear combination $\sum_{i=1}^{m} \lambda_i f_i$ is not degenerate, and $\lambda_1, \ldots, \lambda_m$ can be found in randomized polynomial time. Assuming $\lambda_1 \neq 0$, we substitute the linear combinations $\sum_{i=1}^{m} \lambda_i H_i$ and $\sum_{i=1}^{m} \lambda_i H'_i$ to H_1 and H'_1.

As a consequence, we can find linear forms $\ell_1, \ldots, \ell_n$ in $\mathbf{x}$ such that, see (Lidl and Niederreiter, 1997, Theorem 6.30): $f_1(\mathbf{x}) = \ell_1\ell_2 + \ell_3\ell_4 + \cdots + \ell_{n-1}\ell_n$ or $f_1(\mathbf{x}) = \ell_1\ell_2 + \ell_3\ell_4 + \cdots + \ell_{n-1}\ell_n + \ell_{n-1}^2 + d\ell_n^2$, where $\mathrm{Tr}_{\mathbb{K}}(d) = d + d^2 + \cdots + d^{q/2} = 1$. After applying this change of variables, $\Sigma(H_1)$ is always the following invertible block diagonal matrix:

$$\Sigma(H_1) = \mathrm{Diag}\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right).$$

Following paragraph 2.iv, we once again choose ν_i such that $\Sigma(H_i + \nu_i H_1)$ is invertible and replace H_i by $H_i + \nu_i H_1$. Thus, Proposition 9 and Theorem 10 become:

Proposition 15. *Let n be an even integer and $\mathbb{K}$ be a field of characteristic 2. Let $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[x_1, \ldots, x_n]^m \times \mathbb{K}[x_1, \ldots, x_n]^m$ be regular quadratic homogeneous polynomials. There is a randomized polynomial-time algorithm which returns "*NoSolution*" only if $\mathbf{f} \nsim \mathbf{g}$ or a new instance*

$$(\tilde{\mathbf{f}}, \tilde{\mathbf{g}}) = ((\delta, \tilde{f}_2, \ldots, \tilde{f}_m), (\delta, \tilde{g}_2, \ldots, \tilde{g}_m))\mathbb{K}[x_1, \ldots, x_n]^m \times \mathbb{K}[x_1, \ldots, x_n]^m$$

such that $\mathbf{f} \sim \mathbf{g} \iff \tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$. Furthermore, denoting D the upper triangular matrix of $\tilde{f}_1 = \tilde{g}_1 = \delta$, `IP1S` *comes down to a $\Sigma(D)$-Orthogonal Simultaneous Matrix Conjugacy problem,* i.e. *conjugacy by an $\Sigma(D)$-orthogonal matrix under some constraints:*

$$\begin{aligned} &A^{\mathrm{T}}\Sigma(D)A = \Sigma(D) \text{ and } \forall i, 2 \leq i \leq m, \Sigma(D)^{-1}\Sigma(H'_i) = A^{-1}\Sigma(D)^{-1}\Sigma(H_i)A, \\ &\Delta(A^{\mathrm{T}}H_iA) = \Delta(H'_i). \end{aligned}$$

Proof. We mimic the proof of Proposition 9. We compute in randomized polynomial time $\lambda_1, \ldots, \lambda_m \in \mathbb{K}$ such that $\varphi = \sum_{i=1}^{m} \lambda_i \cdot f_i$ is regular and we define $\gamma = \sum_{i=1}^{m} \lambda_i \cdot g_i$. Assuming w.lo.g. $\lambda_1 \neq 0$. We have then:

$$\mathbf{f} \sim \mathbf{g} \iff (\varphi, f_2, \ldots, f_m) \sim (\gamma, g_2, \ldots, g_m).$$

Computing δ the canonical quadratic form equivalent to φ yields a $P \in \mathrm{GL}_n(\mathbb{K})$ such that $\tilde{\mathbf{f}} = (\tilde{\varphi} = \delta, \tilde{f}_2, \ldots, \tilde{f}_m) = (\varphi(P\mathbf{x}), f_2(P\mathbf{x}), \ldots, f_m(P\mathbf{x}))$.

Then computing the canonical quadratic form equivalent to γ allows us to compare it with δ. If they are different, then $\mathbf{f} \nsim \mathbf{g}$ and we return "NoSolution". Otherwise, the reduction is given by a matrix $Q \in \mathrm{GL}_n(\mathbb{K})$ such that $\tilde{\mathbf{g}} = (\tilde{\gamma} = \delta, \tilde{g}_2, \ldots \tilde{g}_m) = (\gamma(Q\mathbf{x}), g_2(Q\mathbf{x}), \ldots, g_m(Q\mathbf{x}))$. Thus, $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$ if and only if $\mathbf{f} \sim \mathbf{g}$.

Finally, equations (6) $A^{\mathrm{T}}\Sigma(H_i)A = \Sigma(H'_i)$ for all i, $1 \leq i \leq m$ of can be rewritten as $A^{\mathrm{T}}\Sigma(D)A = \Sigma(D)$ and $\Sigma(D)^{-1}\Sigma(H'_i) = A^{-1}\Sigma(D)^{-1}\Sigma(H_i)A$ for all i, $2 \leq i \leq m$, while equations (7) $\Delta(A^{\mathrm{T}}H_iA) = \Delta(H'_i)$ for all i, $1 \leq i \leq m$ remain. □

As a consequence, we designed the following algorithm to solve regular instances of quadratic-IP1S in even dimension over a field of characteristic 2.

Algorithm 2. Probabilistic algorithm in characteristic 2.

Input Two sets of triangular matrices $\mathscr{H} = \{H_1 = D, \ldots, H_m\} \subseteq \mathbb{K}^{n\times n}$ and $\mathscr{H}' = \{H'_1 = D, \ldots, H'_m\} \subseteq \mathbb{K}^{n\times n}$ such that $H_1 + H_1^{\mathrm{T}}$ and $H'_1 + H'^{\mathrm{T}}_1$ are invertible.

Output A description of the matrix $A \in \mathrm{GL}_n(\mathbb{L})$ such that $H'_i = A^{\mathrm{T}} H_i A$ for all $1 \leq i \leq m$ or "NoSolution".

1. Compute the vector subspace $\mathscr{Y} = \{Y \mid \Sigma(D)^{-1}\Sigma(H_i)Y = Y\Sigma(D)^{-1}\Sigma(H'_i),\ \forall 1 \leq i \leq m\} \subseteq \mathbb{K}^{n\times n}$.
2. **If** $\mathscr{Y}$ is reduced to the null matrix **then return** "NoSolution".
3. Pick at random $Y \in \mathscr{Y}$.
4. Compute $Z = \Sigma(D)^{-1}\Sigma(Y)^{\mathrm{T}}\Sigma(D)\Sigma(Y)$ and $J = T^{-1}ZT \in \mathbb{L}^{n\times n}$, the Jordan normal form of Z together with T.
5. **While** J is not diagonal
   a. Pick at random $Y \in \mathscr{Y}$.
   b. Compute Z, J and T as above.
6. Compute G^{-1} the inverse of a square root of J.
7. **Return** YTG^{-1} and T.

The while loop comes from the fact that unlike other characteristics, even if Z is invertible, it might not have any square roots which are polynomials in Z. In Section 6.2, we prove that there exists a square root of Z, which is a polynomial in Z if and only if Z is diagonalizable.

Open Question: The Irregular Case. As stated above, in characteristic 2, the irregular case seems to cover more cases than in other characteristics. Indeed, what is called usually a regular quadratic form in odd dimension falls in the irregular case. However, from the Hessian matrix point of view, an instance is irregular if all linear combinations of the Hessian matrices are singular over the ground field. This allows us to unify our statement about irregularity to all characteristics.

It seems to be an intriguing challenge to solve the binary case on instances with regular quadratic forms, in particular in odd dimension.

3.5. Benchmarks

We present in this section some timings of our algorithms over instances of IP1S. We created instances $\mathscr{H} = \{H_1, \ldots, H_m\}$ and $\mathscr{H}' = \{H'_1, \ldots, H'_m\}$ which are randomly alternatively equivalent over $\mathbb{F}_p$, equivalent over $\mathbb{F}_{p^2}$ but not $\mathbb{F}_p$ or not equivalent at all over $\bar{\mathbb{F}}_p$, the algebraic closure of $\mathbb{F}_p$, for an odd p. We report our timings in the following Table 1 obtained using one core of an Intel Core i7 at 2.6GHz running Magma 2.19, Bosma et al. (1997), on Linux with 16GB of RAM. These timings corresponds to solving the linear system which is the dominant

part in our algorithm with complexity $O(n^{2\omega})$. The code is accessible on the first author's webpage `http://www-polsys.lip6.fr/~berthomieu/IP1S.html`. To simplify the presentation, we only considered the case when $m = n$. That is, we only considered n matrices of size n.

Since our matrices are randomly chosen, we apply the following strategy. We first solve the linear system $H_1^{-1} H_i A = A H_1'^{-1} H_i'$, for all i, $2 \leq i \leq m$. In fact, in practice, $i = 2, 3$ give enough equations to retrieve A up to one free parameter if $\mathscr{H}$ and $\mathscr{H}'$ are indeed equivalent. If the matrices are not equivalent, this linear system will return the zero matrix only.

Then, to determine A, we solve one quadratic equation amongst the ones given by $A^{\mathrm{T}} H_1 A = H_1'$. Let us notice that either all these equations can be solved over $\mathbb{F}_p$ or none of them can. If they can, then $\mathscr{H}$ and $\mathscr{H}'$ are equivalent over $\mathbb{F}_p$ and we have determined A up to a sign, otherwise $\mathscr{H}$ and $\mathscr{H}'$ are only equivalent over $\mathbb{F}_{p^2}$ but not $\mathbb{F}_p$ and we also have computed such an A. This yields Algorithm 3.

Algorithm 3. Simplified Algorithm.

Input Two sets of generic invertible symmetric matrices $\mathscr{H} = \{H_1, \ldots, H_m\} \subseteq \mathbb{K}^{n \times n}$ and $\mathscr{H}' = \{H_1', \ldots, H_m'\} \subseteq \mathbb{K}^{n \times n}$.

Output A matrix $A \in \mathrm{GL}_n(\mathbb{K})$ such that $H_i' = A^{\mathrm{T}} H_i A$ for all $1 \leq i \leq m$ or "NoSolution".

1. Compute the vector subspace $\mathscr{Y} = \{Y \mid H_1^{-1} H_i Y = Y H_1'^{-1} H_i', \forall 2 \leq i \leq 3\} \subseteq \mathbb{K}^{n \times n}$.
2. **If** $\mathscr{Y}$ is reduced to a space of singular matrices **then return** "NoSolution".
3. Determine Y_0 such that $\mathscr{Y} = \{\lambda Y_0 \mid \lambda \in \mathbb{K}\}$.
4. Solve in λ one equation $\lambda^2 (Y_0^{\mathrm{T}} H_1 Y_0)_{i,j} = (H_1')_{i,j}$ for a suitable pair (i, j).
5. Set $A = \lambda Y_0$.
6. Pick at random $r \in \mathbb{K}^n$.
7. Check that $A^{\mathrm{T}} H_i A r = H_i' r$ for all i, $1 \leq i \leq m$.
8. **Return** A.

Complexity estimate. Taking the first 3 matrix equations $H_1^{-1} H_i Y = Y H_1'^{-1} H_i'$, in the n^2 unknowns, one can solve this system in $O(n^{2\omega})$ operations in $\mathbb{K}$. Then, one needs to determine λ by extracting one square root in $\mathbb{K}$ which can be done in $O((\log q)^3)$ operations in $\mathbb{K} = \mathbb{F}_q$ with Tonelli–Shanks's algorithm, Shanks (1973). Finally, one can check that $A^{\mathrm{T}} H_i A = H_i'$ for all i, with high probability, by picking up at random a vector and checking that the products of this vector with both sets of matrices coincides. This can be done in $O(mn^2)$ operations in $\mathbb{K}$.

Recall that, in practice, the best matrix multiplication algorithm is due to Strassen (1969) whose complexity is in $O(n^{\log_2 7}) \subseteq O(n^{2.807})$. Thus, our complexity is in $O(n^{5.615})$. This complexity is well confirmed since multiplying by 2 the sizes and the number of matrices multiplies our timings roughly by at most 50.

In Table 2, we report our timings for solving the linear system of our algorithm in characteristic 2 presented in Section 3.4. Our method does not differ much from the one in odd characteristic. We pick at random two sets of m upper triangular matrices over $\mathbb{F}_2$ which are either equivalent over

n	20	30	40	50	60	70	80	90	100
Timings	0.040	0.20	0.84	2.7	7.5	17	40	79	130

Table 1: Timings for solving `IP1S` over $\mathbb{F}_{65521}$ in s.

n	20	30	40	50	60	70	80	90	100
Timings (MAGMA)	0.010	0.030	0.080	0.25	0.68	1.4	3.2	6.25	16
Timings (M4RI)			0.010	0.030	0.06	0.14	0.27	0.51	0.91

Table 2: Timings for solving `IP1S` over $\mathbb{F}_2$ in s.

$\mathbb{F}_2$ or not equivalent at all over $\bar{\mathbb{F}}_2$, the algebraic closure of $\mathbb{F}_2$. We first solve the linear system $\Sigma(H_1)^{-1}\Sigma(H_i)A = A\Sigma(H'_1)^{-1}\Sigma(H'_i)$, for all i, $2 \leq i \leq m$. Let us notice that in dimension 2, the linear system does not yield any information on A. In dimensions 8 or more (resp. 4 and 6), if $\mathscr{H}$ and $\mathscr{H}'$ are equivalent, the linear system yields in general A up to one free parameter if $m \geq 3$ (resp. $m \geq 5$). Otherwise, it yields the zero matrix. Then, it suffices to solve one of the quadratic equations amongst the one given by $A^{\mathrm{T}}\Sigma(H_1)A = \Sigma(H'_1)$ and $\Delta(A^{\mathrm{T}}H_iA) = \Delta(H'_i)$, for all i, $1 \leq i \leq m$ (see Proposition 15).

We compare the timings of both MAGMA and the C library M4RI, due to Albrecht and Bard (2012).

Once again, our complexity in $O(n^{2\omega})$ is well confirmed by our timings. Thanks to the linear system which totally determines A up to one free parameter, we just need to set this parameter to 1 to obtain A. This also explains why our timings are better than over $\mathbb{F}_{65521}$ although it would seem a lot of quadratic equations must be solved.

4. Counting the Solutions: #`IP1S`

In this part, we present a method for counting the number of solutions to quadratic-`IP1S`. The main result is a consequence of (Singla, 2010, Lemma 4.11). According to Proposition 5, this is equivalent to enumerating all the invertible linear transformations on the variables between two sets of quadratic homogeneous polynomials. We provide here an upper bound on the number of solutions. We consider in this part regular quadratic homogeneous instances $(\mathbf{f},\mathbf{g}) \in \mathbb{K}[\mathbf{x}]^m \times \mathbb{K}[\mathbf{x}]^m$.

Let $\mathscr{H} = \{H_1 = D, \ldots, H_m\}$ and $\mathscr{H}' = \{H'_1 = D, \ldots, H'_m\}$ be the Hessian matrices in $\mathbb{K}^{n\times n}$ of $\mathbf{f}$ and $\mathbf{g}$ respectively. Our counting problem is equivalent to enumerating the number of D-orthogonal matrices X satisfying:

$$X^{-1}D^{-1}H_iX = D^{-1}H'_i, \quad \forall i, 1 \leq i \leq m. \tag{8}$$

In (Singla, 2010, Section 4), the author computes the set of all matrices commuting with a given matrix. In particular, from Lemma 4.11 of the aforementioned paper, we can determine the size of this set and thus our upper bound on the number of solutions to quadratic-`IP1S`. In order to be self-contained, the proofs of the following lemmas shall be found in Appendix A.

Let us notice that if X and X' are both orthogonal solutions of (8), then XX'^{-1} commutes with $D^{-1}\mathscr{H}$ (resp. $X^{-1}X'$ commutes with $D^{-1}\mathscr{H}'$). Therefore, the size of the set of solutions is upper bounded by the number of invertible elements in the centralizer $\mathscr{C}(D^{-1}\mathscr{H})$ of $D^{-1}\mathscr{H}$.

Let α be an algebraic element of degree m over $\mathbb{K}$ and let $\mathbb{K}' = \mathbb{K}(\alpha)$. We consider the matrix $H = D^{-1}\left(H_1 + \cdots + \alpha^{m-1}H_m\right) \in \mathbb{K}'^{n\times n}$. It is clear that a matrix $X \in \mathbb{K}^{n\times n}$ is such that $X^{-1}D^{-1}H_iX = D^{-1}H_i$ for all $i, 1 \leq i \leq m$ if and only if $X^{-1}HX = H$. Hence, the problem again reduces itself to the computation of the centralizer $\mathscr{C}(H)$ of H intersected with $\mathrm{GL}_n(\mathbb{K})$. To ease the analysis, we consider the subspace $\mathscr{V} = \mathscr{C}(H) \cap \mathbb{K}^{n\times n}$ of matrices in $\mathbb{K}^{n\times n}$ commuting with H. This provides an upper bound on the number of solutions. The dimension of $\mathscr{V}$ as a $\mathbb{K}$-vector space is upper bounded by the dimension of $\mathscr{C}(H)$ as a $\mathbb{K}'$-vector space. Indeed, $\mathscr{V} \otimes \mathbb{K}' \subseteq \mathscr{C}(H)$, hence $\dim_{\mathbb{K}} \mathscr{V} = \dim_{\mathbb{K}'}(\mathscr{V} \otimes \mathbb{K}') \leq \dim_{\mathbb{K}'} \mathscr{C}(H)$. Since we only want the size of the centralizer of H, we can restrict our attention to the centralizer of the Jordan normal form J of H defined over a field $\mathbb{L}$.

Let us denote $\zeta_1, \ldots, \zeta_r$ the eigenvalues of J. According to (Singla, 2010, Lemma 4.11) and Lemma 23 in Appendix A, if J is made of Jordan blocks of size $s_{i,1} \leq \cdots \leq s_{i,d_i}$ for $i, 1 \leq i \leq r$, then centralizer of H has dimension at most

$$\sum_{1\leq i\leq r} \sum_{1\leq j\leq d_i} (2d_i - 2j + 1)s_{i,j}.$$

As a consequence, if q is an odd prime power, then the following corollary gives an upper bound on the number of solutions of quadratic-`IP1S` in $\mathbb{F}_q^{n\times n}$.

Corollary 16. *Let $H_1, \ldots, H_m \in \mathbb{F}_q^{n\times n}$ be symmetric matrices. Let α be algebraic over $\mathbb{F}_q$ of degree m. Let $H = \sum_{i=1}^m \alpha^{i-1}H_i \in \mathbb{F}_{q^m}^{n\times n}$ and let J be its normal Jordan form with eigenvalues $\zeta_1, \ldots, \zeta_r$. Assuming the blocks of J associated to ζ_i are $J_{\zeta_i,s_{i,1}}, \ldots, J_{\zeta_i,s_{i,d_1}}$ with $s_{i,1} \leq \cdots \leq s_{i,d_i}$ for $i, 1 \leq i \leq r$, then the number of solutions of quadratic-`IP1S` in $\mathbb{F}_q^{n\times n}$ on the instance $(H_1, \ldots, H_m)$ is at most*

$$q^{\left(\sum_{1\leq i\leq r}\sum_{1\leq j\leq d_i}(2d_i-2j+1)s_{i,j}\right)} - 1.$$

As mentioned in the introduction, the counting problem considered here is related to cryptographic concerns. It corresponds to evaluating the number of equivalent secret keys in `MPKC` (see Faugère et al. (2012); Wolf and Preneel (2011)). In particular, in Faugère et al. (2012), the authors propose an "ad-hoc" method for solving a particular instance of `#IP1S`. An interesting open question would be to revisit the results from Faugère et al. (2012) with our approach.

5. Special Case of the general IP Problem

In this part, we present a randomized polynomial-time algorithm for the following task:
Input: $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{K}[x_1, \ldots, x_n]^n$, and $\mathbf{POW}_{n,d} = (x_1^d, \ldots, x_n^d) \in \mathbb{K}[x_1, \ldots, x_n]^n$ for some $d > 0$.
Question: Find – if any – $(A, B) \in \mathrm{GL}_n(\mathbb{K}) \times \mathrm{GL}_n(\mathbb{K})$ such that:

$$\mathbf{g} = B \cdot \mathbf{POW}_{n,d}(A \cdot \mathbf{x}), \text{ with } \mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}}.$$

In Kayal (2011), the author proposes a randomized polynomial-time algorithm for solving the problem when B is the identity matrix and $m = 1$ of finding A such that $g(\mathbf{x}) = f(A \cdot \mathbf{x})$ with $f(\mathbf{x}) = \sum_{i=1}^{n} x_i^d$.

We generalize this result to $m = n$ with an additional transformation on the polynomials. The main tool of our method is the following theorem.

Theorem 17. *Let $\mathbf{g} = (g_1, \ldots, g_n)$ be polynomials of degree d over $\mathbb{K}[x_1, \ldots, x_n]$ given in dense representation. Let L be a polynomial size of an arithmetic circuit to evaluate the determinant of the Jacobian matrix of $\mathbf{g}$. If the size of $\mathbb{K}$ is at least $12 \max(2^{L+2}, d(n-1)2^{d(n-1)} + d^3(n-1)^3, 2(d(n-1)+1)^4)$, then one can factor the determinant of the Jacobian matrix of $\mathbf{g}$ in randomized polynomial time.*

Proof. For this, we will use Kaltofen (1989)'s algorithm to factor a polynomial given by evaluation, the needed size of $\mathbb{K}$ is a consequence of this. As $\mathbf{g}$ has at most $n\binom{n+d-1}{d} \in O(n^{d+1})$ monomials, it can be evaluated in polynomial time using a multivariate Horner's scheme. Each $\frac{\partial g_i}{\partial x_j}(\mathbf{a})$ is obtained as the coefficient in front of x_j of the expansion of $g_i(a_1, \ldots, a_{j-1}, a_j + x_j, a_{j+1}, \ldots, a_n)$ which is a univariate polynomial of degree at most d. By (Bini and Pan, 1994, Chapter 1, Section 8), this can be computed as the shift of a polynomial in polynomial time. Hence the Jacobian matrix of $\mathbf{g}$ at $\mathbf{a}$ can be evaluated in polynomial time with an arithmetic circuit of polynomial size L. This circuit can be for instance the evaluation of the Jacobian matrix of $\mathbf{g}$ followed by a Gaussian elimination on the matrix to compute the determinant. The determinant of the matrix can be recovered by linear algebra in $O(n^\omega)$ operations, with ω being the exponent of time complexity of matrix multiplication, $2 \leq \omega \leq 3$. Using the arithmetic circuit of polynomial size L to evaluate the determinant of the Jacobian matrix, one can use KALTOFEN's algorithm to factor it in polynomial time. □

As in Kayal (2011) or Perret (2005), we use partial derivatives to extract matrices A and B. The idea is to factor the Jacobian matrix (whereas Kayal (2011) uses the Hessian matrix) of $\mathbf{g}$ at $\mathbf{x}$ which is defined as follows:

$$\mathrm{J}_{\mathbf{g}}(\mathbf{x}) = \left(\partial_j g_i = \frac{\partial g_i}{\partial x_j} \right)_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}.$$

According to the following lemma, the Jacobian matrix is especially useful in our context:

Lemma 18. *Let $\big(\mathbf{f} = (f_1, \ldots, f_m), \mathbf{g} = (g_1, \ldots, g_m)\big) \in \mathbb{K}[x_1, \ldots, x_n]^m \times \mathbb{K}[x_1, \ldots, x_n]^m$. If $(A, B) \in \mathrm{GL}_n(\mathbb{K}) \times \mathrm{GL}_m(\mathbb{K})$ are such that $\mathbf{g} = B \cdot \mathbf{f}(A \cdot \mathbf{x})$, then*

$$\mathrm{J}_{\mathbf{g}}(\mathbf{x}) = B \cdot \mathrm{J}_{\mathbf{f}}(A \cdot \mathbf{x}) \cdot A.$$

As a consequence, $\det \mathrm{J}_{\mathbf{g}}(\mathbf{x}) = \det A \cdot \det B \cdot \det \mathrm{J}_{\mathbf{f}}(A \cdot \mathbf{x})$.

As long as $\mathrm{char}\,\mathbb{K}$ does not divide d, the Jacobian matrix of $\mathbf{f} = \mathbf{POW}_{n,d}(\mathbf{x})$ is an invertible diagonal matrix whose diagonal elements are $\big(\mathrm{J}_{\mathbf{f}}(\mathbf{x})\big)_{i,i} = d \cdot x_i^{d-1}$, $\forall i, 1 \leq i \leq n$. Thus:

$$\det \mathrm{J}_{\mathbf{POW}_{n,d}}(\mathbf{x}) = d^n \prod_{i=1}^{n} x_i^{d-1}.$$

This gives

Lemma 19. *Let $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{K}[x_1, \ldots, x_n]^n$. Let $d > 0$ be an integer, and define $\mathbf{POW}_{n,d} = (x_1^d, \ldots, x_n^d) \in \mathbb{K}[x_1, \ldots, x_n]^m$. If $(A, B) \in \mathrm{GL}_n(\mathbb{K}) \times \mathrm{GL}_n(\mathbb{K})$ are such that $\mathbf{g}(\mathbf{x}) = B \cdot \mathbf{POW}_{n,d}(A \cdot \mathbf{x})$, then:*

$$\det \mathrm{J}_{\mathbf{g}}(\mathbf{x}) = c \cdot \prod_{i=1}^{n} \ell_i(\mathbf{x})^{d-1},$$

with $c \in \mathbb{K} \setminus \{0\}$, and the ℓ_i's are linear forms whose coefficients are the ith rows of A.

Proof. According to Lemma 18, $\det\left(\mathrm{J}_{\mathbf{g}}(\mathbf{x})\right) = \det(A) \cdot \det(B) \cdot d^n \cdot \prod_{i=1}^{n} \ell_i(\mathbf{x})^{d-1}$. □

From Lemmas 19, we can derive a randomized polynomial-time algorithm for solving IP on the instance $(\mathbf{f} = \mathbf{POW}_{n,d}, \mathbf{g}) \in \mathbb{K}[x_1, \ldots, x_n]^n \times \mathbb{K}[x_1, \ldots, x_n]^n$ in characteristic 0. It suffices to use Kaltofen (1989)'s algorithm for factoring $\det \mathrm{J}_{\mathbf{g}}(\mathbf{x})$ in randomized polynomial time.

This allows us to recover – if any – the change of variables A. The matrix B can be then recovered by linear algebra, *i.e.* solving a linear system of equations. This proves the result announced in the introduction for IP, that is Theorem 4 whenever $\mathrm{char}\,\mathbb{K} \nmid d$.

Small characteristic. If $\mathrm{char}\,\mathbb{K}$ divides d, we must change a little bit our strategy. Let us write $d = p^r e$ with $\mathrm{char}\,\mathbb{K} = p$ and e coprime. Then,

$$\begin{aligned}
\mathbf{POW}_{n,d}(A\,\mathbf{x}) &= \left(\left(\sum_{j=1}^{n} a_{1,j} x_j \right)^{p^r e}, \ldots, \left(\sum_{j=1}^{n} a_{n,j} x_j \right)^{p^r e} \right) \\
\mathbf{POW}_{n,d}(A\,\mathbf{x}) &= \left(\left(\sum_{j=1}^{n} a_{1,j}^{p^r} x_j^{p^r} \right)^{e}, \ldots, \left(\sum_{j=1}^{n} a_{n,j}^{p^r} x_j^{p^r} \right)^{e} \right) \\
\mathbf{POW}_{n,d}(A\,\mathbf{x}) &= \mathbf{POW}_{n,e}\left(A^{(p^r)}\,\mathbf{x}^{p^r} \right),
\end{aligned}$$

with $A^{(p^r)} = \left(a_{i,j}^{p^r} \right)_{1 \le i,j \le n}$ and $\mathbf{x}^{p^r} = \left(x_1^{p^r}, \ldots, x_n^{p^r} \right)$. Thus $\mathbf{g}$ is a polynomial in $\mathbf{x}^{p^r}$ and by replacing $\mathbf{x}^{p^r}$ by $\mathbf{x}$, the problem comes down to checking if $\tilde{\mathbf{g}} = B \cdot \mathbf{POW}_{n,e}(A^{(p^r)} \cdot \mathbf{x})$ where $\mathbf{g}(\mathbf{x}) = \tilde{\mathbf{g}}(\mathbf{x}^{p^r})$. Now, as $\tilde{\mathbf{g}}(\mathbf{x}) = B \cdot \mathbf{POW}_{n,e}\left(A^{(p^r)}\,\mathbf{x}\right)$, then

$$\mathrm{J}_{\tilde{\mathbf{g}}} = B \cdot \mathrm{J}_{\mathbf{POW}_{n,e}}(A^{(p^r)}\,\mathbf{x}) \cdot A^{(p^r)}.$$

Hence, $\det \mathrm{J}_{\tilde{\mathbf{g}}}(\mathbf{x}) = \det B \det \mathrm{J}_{\mathbf{POW}_{n,e}}(A^{(p^r)}\,\mathbf{x}) \det A^{(p^r)} = e^n \prod_{i=1}^{n} \tilde{\ell}_i(\mathbf{x})^{e-1} (\det A)^{p^r} \det B$, where the $\tilde{\ell}_i$'s are linear forms whose coefficients are the ith rows of $A^{(p^r)}$. Then, to use KALTOFEN's algorithm, one must set a low enough probability of failure ε yielding a big enough set of sampling points, see (Kaltofen, 1989, Section 6, Algorithm, Step R). In particular, if the arithmetic circuit for evaluating the determinant of the Jacobian we want to factor has size L, then the size of the sampling set must be greater than

$$\frac{6}{\varepsilon} \max\left(2^{L+2}, e(n-1)\,2^{e(n-1)} + e^3\,(n-1)^3, 2\,(e(n-1)+1)^4 \right),$$

recalling that our polynomial has degree $e(n-1)$. In other words, if the probability of failure is less than $1/2$, then one must consider a field of size at least

$$12\max\left(2^{L+2}, e(n-1)\,2^{e(n-1)} + e^3\,(n-1)^3, 2\,(e(n-1)+1)^4\right).$$

All in all, this allows us to retrieve – if any – the change of variables $A^{(p^r)}$ and thus A. Then B can be recovered by linear algebra. This proves Theorem 4 for any characteristic, as in the introduction.

Theorem 4 (restated). *Let $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{K}[x_1, \ldots, x_n]^n$ be given in dense representation, and $\mathbf{f} = \mathbf{POW}_{n,d} = (x_1^d, \ldots, x_n^d) \in \mathbb{K}[x_1, \ldots, x_n]^n$ for some $d > 0$. Whenever $\operatorname{char}\mathbb{K} = 0$, let $e = d$ and $\tilde{\mathbf{g}} = \mathbf{g}$. Otherwise, let $p = \operatorname{char}\mathbb{K}$, let e and r be integers such that $d = p^r e$, p and e coprime, and let $\tilde{\mathbf{g}} \in \mathbb{K}[\mathbf{x}]^n$ be such that $\mathbf{g}(\mathbf{x}) = \tilde{\mathbf{g}}(\mathbf{x}^{p^r})$. Let L be a polynomial size of an arithmetic circuit to evaluate the determinant of the Jacobian matrix of $\tilde{\mathbf{g}}$. If the size of $\mathbb{K}$ is at least $12\max(2^{L+2}, e(n-1)\,2^{e(n-1)} + e^3\,(n-1)^3, 2\,(e(n-1)+1)^4)$, then there is a randomize polynomial-time algorithm which recovers – if any – $(A,B) \in \mathrm{GL}_n(\mathbb{K}) \times \mathrm{GL}_n(\mathbb{K})$ such that:*

$$\mathbf{g} = B \cdot \mathbf{POW}_{n,d}(A \cdot \mathbf{x}).$$

The computation of such a pair (A,B) is summarized in the following Algorithm.

Algorithm 4. `IP` for $\mathbf{POW}_{n,d}$ and $\mathbf{g}$.

Input One set of polynomials $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{K}[x_1, \ldots, x_n]^n$, homogeneous of degree d.

Output Two matrices $A, B \in \mathrm{GL}_n(\mathbb{K})$ – if any – such that $\mathbf{g}(\mathbf{x}) = B \cdot \mathbf{POW}_{n,d}(A \cdot \mathbf{x})$ with $\mathbf{POW}_{n,d}(\mathbf{x}) = (x_1^d, \ldots, x_n^d)$ or "NoSolution".

1. **If** $\operatorname{char}\mathbb{K} = p > 0$ **then**
   a. Compute r, e such that $d = p^r e$ with p and e coprime.
   b. Compute $\tilde{\mathbf{g}}$ such that $\tilde{\mathbf{g}}(\mathbf{x}^{p^r}) = \mathbf{g}(\mathbf{x})$.
2. **Else** $e = d$ and $\tilde{\mathbf{g}} = \mathbf{g}$.
3. Create an arithmetic circuit of polynomial size L to evaluate $\det \mathrm{J}_{\mathbf{g}}(\mathbf{x})$.
4. Evaluate $\det \mathrm{J}_{\mathbf{g}}(\mathbf{x})$ in at least $12\max(2^{L+2}, e(n-1)\,2^{e(n-1)} + e^3\,(n-1)^3, 2\,(e(n-1)+1)^4)$ distinct points.
5. Factor $\det \mathrm{J}_{\mathbf{g}}(\mathbf{x})$ with Kaltofen's algorithm.
6. **If** the factorization is $c \prod_{i=1}^n \ell_i(\mathbf{x})^{e-1}$ **then** $A = \left(\ell_{i,j}^{e/d}\right)_{1 \le i,j \le n}$.
7. **Else return** "NoSolution".
8. Compute B such that $\tilde{\mathbf{g}}(\mathbf{x}) = B \cdot (\ell_1(\mathbf{x})^e, \ldots, \ell_n(\mathbf{x})^e)^{\mathrm{T}}$.
9. **Return** (A,B).

6. Square Root of a Matrix

In this section, we present further algorithms for computing the square root of a matrix. We use the same notation as in Section 3. A square root of a matrix Z is a matrix whose square is Z. In the first subsection, we deal with some properties of the square root of a matrix in characteristic not 2. In particular, we show that an invertible matrix Z always has a square root which is a polynomial in Z. In the second subsection, we consider the case of characteristic 2. We recall that whenever Z is not diagonalizable, then Z might have a square root but it is never a polynomial in Z. We give some examples of such matrices Z. Lastly, we propose an alternative to the method of Section 3 for computing the square root of a matrix in polynomial time for any field of characteristic $p \geq 2$.

6.1. The square root as a polynomial in characteristic not 2

In this part, we prove that an invertible matrix always has a square root which is a polynomial in considered matrix. More specifically, we shall prove the following result.

Proposition 20. *Let $Z \in \mathbb{K}^{n\times n}$ be an invertible matrix whose eigenvalues are $\zeta_1, \ldots, \zeta_r$. Let $\omega_1, \ldots, \omega_r$ be such that $\omega_i^2 = \zeta_i$ and $\zeta_i = \zeta_j \Rightarrow \omega_i = \omega_j$, for all $1 \leq i, j \leq r$. Then, there exists $W \in \mathbb{K}(\omega_1, \ldots, \omega_r)[Z]$ a square root of Z whose eigenvalues are $\omega_1, \ldots, \omega_r$.*

Proof. Let T be a matrix of change of basis, such that $J = T^{-1}ZT$ is made of Jordan blocks. It is clear that W such that $W^2 = Z$ is a polynomial in Z, *i.e.* $W = Q(Z)$, if and only if $G = T^{-1}WT$ satisfies $G = Q(J)$. Let $J_{\zeta,d}$ be the Jordan block of size d associated with eigenvalue ζ and ω be a square root of ζ. We shall first prove that the square root $G_{\omega,d}$ of $J_{\zeta,d}$ is a polynomial in $J_{\zeta,d}$ with coefficients in $\mathbb{K}(\omega)$. Matrix $J_{\zeta,d} - \zeta\,\mathrm{Id}_d$ is nilpotent of degree d. Hence, by the classical Taylor expansion of the square root near Id_d, one can write

$$G_{\omega,d} = \omega \sum_{k=0}^{d-1} \binom{1/2}{k} \zeta^{-k} \left(J_{\zeta,d} - \zeta\,\mathrm{Id}\right)^k = \sum_{k=0}^{d-1} \binom{1/2}{k} \omega^{1-2k} \left(J_{\zeta,d} - \zeta\,\mathrm{Id}\right)^k = Q_\zeta(J_{\zeta,d}) \qquad (9)$$
$$= \begin{pmatrix} \omega & \binom{1/2}{1}\omega^{-1} & \cdots & \binom{1/2}{d-1}\omega^{3-2d} \\ & \ddots & \ddots & \vdots \\ & & \ddots & \binom{1/2}{1}\omega^{-1} \\ & & & \omega \end{pmatrix},$$

with $Q_\zeta(x) = \sum_{k=0}^{d-1} \binom{1/2}{k} \omega^{1-2k} (x-\zeta)^i \in \mathbb{K}(\omega)[x]$.

It remains to prove that for multiple Jordan blocks, one can find a common polynomial. From equation (9), we deduce that G is a polynomial in $J = \mathrm{Diag}\left(J_{\zeta_1,d_1}, \ldots, J_{\zeta_r,d_r}\right)$ if and only if there exists a polynomial Q such that $Q = Q_{\zeta_i} \mod (X-\zeta_i)^{d_i}$, for all i, $1 \leq i \leq r$. By the Chinese Remainder Theorem, this can always be solved as soon as $\zeta_i = \zeta_j$ implies $Q_{\zeta_i} = Q_{\zeta_j} \mod (X - \zeta_i)^{\min(d_i,d_j)}$, which is exactly the condition $\omega_i = \omega_j$. □

Let us notice that picking the same square root for two equal eigenvalues is necessary. Indeed, although $W = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ is a square root of $Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, $W \notin \mathbb{K}[Z]$.

6.2. Matrices with square roots in characteristic 2

In this part, we consider the trickier case of computing the square root of a matrix over a field $\mathbb{K}$ with $\operatorname{char}\mathbb{K} = 2$. Unfortunately, unlike other characteristics, an invertible matrix has not necessarily a square root over $\bar{\mathbb{K}}$. In fact, no Jordan block of size at least 2 has any square root. This is mainly coming from the fact that generalized binomial coefficients $\binom{1/2}{k}$, involved in the Taylor expansion, are meaningless in characteristic 2.

Proposition 21. *Let $Z \in \mathbb{K}^{n\times n}$ be a Jordan normal form with blocks $J_1,\dots,J_r$ of sizes $d_1,\dots,d_r \geq 2$, associated to eigenvalues $\zeta_1,\dots,\zeta_r$ and blocks of sizes* 1 *with eigenvalues $\upsilon_1,\dots,\upsilon_s$. We assume that $J_1,\dots,J_r$ are ordered by decreasing sizes and then eigenvalues. Matrix Z has a square root W if and only if $d_1 - d_2 \leq 1$ and $\zeta_1 = \zeta_2$, $d_3 - d_4 \leq 1$ and $\zeta_3 = \zeta_4$,* etc. *and if for each J_i of size* 2 *that is not paired with J_{i-1} or J_{i+1}, then there exists a j such that $\upsilon_j = \zeta_i$.*

Furthermore, matrix W is a polynomial in Z if and only if Z is diagonalizable.

Before, proving this result, we give some example of matrices with or without square roots. Following matrices J and J' both have two Jordan blocks associated with eigenvalue ζ. Denoting ω the square root of ζ, then K is the square root of J and K'_1, K'_2 are those of J' for x, y, z any.

$$J = \begin{pmatrix} \zeta & 0 & 0 \\ 0 & \zeta & 1 \\ 0 & 0 & \zeta \end{pmatrix}, \quad K = \begin{pmatrix} \omega & 0 & x \\ \frac{1}{x} & \omega & y \\ 0 & 0 & \omega \end{pmatrix},$$

$$J' = \begin{pmatrix} \zeta & 1 & 0 & 0 \\ 0 & \zeta & 0 & 0 \\ 0 & 0 & \zeta & 1 \\ 0 & 0 & 0 & \zeta \end{pmatrix}, \quad K'_1 = \begin{pmatrix} \omega & x & 0 & y \\ 0 & \omega & 0 & 0 \\ \frac{1}{y} & z & \omega & x \\ 0 & \frac{1}{y} & 0 & \omega \end{pmatrix}, \quad K'_2 = \begin{pmatrix} \omega & x & y & z \\ 0 & \omega & 0 & y \\ 0 & \frac{1}{y} & \omega & x \\ 0 & 0 & 0 & \omega \end{pmatrix}.$$

As one can see, none of K, K'_1 and K'_2 are polynomials in J or J' because of the nonzero subdiagonal elements $1/x$ and $1/y$. Examples of matrices without square roots are J'', with two Jordan blocks associated with ζ of sizes 1 and 3, and J''', with three Jordan blocks associated with ζ of size 2. Computing a square root of each of them yields an inconsistent system.

$$J'' = \begin{pmatrix} \zeta & 0 & 0 & 0 \\ 0 & \zeta & 1 & 0 \\ 0 & 0 & \zeta & 1 \\ 0 & 0 & 0 & \zeta \end{pmatrix}, \quad J''' = \operatorname{Diag}\left(\begin{pmatrix} \zeta & 1 \\ 0 & \zeta \end{pmatrix}, \begin{pmatrix} \zeta & 1 \\ 0 & \zeta \end{pmatrix}, \begin{pmatrix} \zeta & 1 \\ 0 & \zeta \end{pmatrix} \right).$$

Proof. Let J be a Jordan block of size d associated to eigenvalue ζ. Then $J^2 - \zeta^2\mathrm{Id} = \begin{pmatrix} \mathbf{0} & \mathrm{Id}_{d-2} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ and one can deduce that ζ^2 is the sole eigenvalue of J^2 but that its geometric multiplicity is 2. Hence the Jordan normal form of J^2 is made of two Jordan blocks.

As $(J-\zeta\mathrm{Id})^d = 0$ and $(J-\zeta\mathrm{Id})^e \neq 0$ for all $e < d$, then $\left(J^2 - \zeta^2\mathrm{Id}\right)^{\lceil d/2 \rceil} = 0$ and $\left(J^2 - \zeta^2\mathrm{Id}\right)^e \neq 0$ for $e < \lceil d/2 \rceil$, *i.e.* $e < d/2$ if d is even and $e < (d+1)/2$ if d is odd. Thus the Jordan normal

form of J^2 has a block of size exactly $\lceil d/2 \rceil$. That is, if d is even, both blocks have size $d/2$ and if d is odd, one block has size $(d+1)/2$ and the other block has size $(d-1)/2$.

By this result, if Z is a square, then one must be able to pair up its Jordan blocks with same eigenvalue ζ so that the sizes differ by at most 1. The blocks that need not be paired being the blocks of size 1.

Conversely, assuming one can pair up the Jordan blocks of Z with same eigenvalue ζ so that the sizes differ by at most 1 and the remaining blocks have sizes 1. Then, each pair of blocks is the Jordan normal form of the square of a Jordan block of size the sum of the sizes and eigenvalue $\sqrt{\zeta}$. Furthermore, each lonely block of size 1 associated with ζ is the square of the block of size 1 associated with $\sqrt{\zeta}$.

Finally, for the last statement, the if part is easy. It remains the only if part for which we assume $W^2 = Z$ and Z is not diagonalizable. Let J be the Jordan normal form of Z with blocks $J_1, \ldots, J_r$. For any polynomial P, $P(J)$ is also block diagonal with blocks $P(J_1), \ldots, P(J_r)$. Thus, if $P(J)^2 = J$, then $P(J_i)^2 = J_i$ for all $1 \leq i \leq r$, which is false, unless J_i has size 1. □

6.3. Computation in characteristic $p \geq 2$

In this part, we present an alternative method to the one presented in Section 3.2. We aim at diminishing the number of variables needed in the expression of the square root. However, this method does not work in characteristic 0. For the time being, we consider $\mathrm{char}\,\mathbb{K} > 2$. However, we shall see below how to adapt this method to the characteristic 2.

The idea is still to perform a change of basis T over $\mathbb{K}$ so that $J = T^{-1}ZT$ has an easily computable square root. This matrix J is the *generalized Jordan normal form*, also known as the *primary rational canonical form* of Z. As the classical Jordan normal form, if Z is diagonalizable over $\bar{\mathbb{K}}$, then J is block diagonal, otherwise it is a block upper triangular matrix. Its diagonal blocks are companion matrices $\mathscr{C}(P_1), \ldots, \mathscr{C}(P_r)$ of irreducible factors $P_1, \ldots, P_r$ of its characteristic polynomial. Superdiagonal blocks are zero matrices with eventually a 1 on the bottom-left corner, if the geometric multiplicity associated to the roots of one the P_i is not large enough. In other words, it gathers d conjugated eigenvalues in one block of size d which is the companion matrix of their shared minimal polynomial. Let us note that computing such a normal form can be done in polynomial time and that the change of basis matrix T is defined over $\mathbb{K}$, see Matthews (1992); Storjohann (1998). Thus, after computing a square root G of J, one can retrieve W and A of Section 3.2 in $O(n^\omega)$ operations in the field of coefficients of G, with ω being the exponent of the time-complexity of matrix multiplication $2 \leq \omega \leq 3$. Furthermore, computing a square root of J is equivalent to computing the square root of each companion matrix. Finally, using the same argument as for the more classical Jordan normal form in Section 6.1, G is a polynomial in J. In the following, we only show how to determine the square root of a companion matrix $\mathscr{C}(P)$, for an irreducible P.

Let $P = x^d + p_{d-1}x^{d-1} + \cdots + p_0$, let us recall that the companion matrix of P is

$$\mathscr{C}(P) = \begin{pmatrix} 0 & & & -p_0 \\ 1 & \ddots & & -p_1 \\ & \ddots & 0 & \vdots \\ & & 1 & -p_{d-1} \end{pmatrix}.$$

If polynomial P can be decomposed as $P(z) = (z-\alpha_0)\cdots(z-\alpha_{d-1})$, then we want to find a polynomial Q such that $Q(z) = (z-\beta_0)\cdots(z-\beta_{d-1})$, where $\beta_i^2 = \alpha_i$ for all $0 \leq i \leq d-1$. Let us notice that

$$P(z^2) = (z^2-\alpha_0)\cdots(z^2-\alpha_{d-1}) = Q(z)\,(z+\beta_0)\cdots(z+\beta_{d-1}) = (-1)^d\,Q(z)\,Q(-z).$$

As a consequence, the characteristic polynomial of $\mathscr{C}(Q)^2$ is

$$\det(\lambda \mathrm{Id} - \mathscr{C}(Q)^2) = \det(\sqrt{\lambda}\mathrm{Id} - \mathscr{C}(Q))\det(\sqrt{\lambda}\mathrm{Id} + \mathscr{C}(Q)) = (-1)^d Q(\sqrt{\lambda})Q(-\sqrt{\lambda}) = P(\lambda).$$

But since P is irreducible over $\mathbb{K}$, by the invariant factors theory, then $\mathscr{C}(Q)^2$ must be similar to the companion matrix $\mathscr{C}(P)$ over $\mathbb{K}$.

As P is irreducible over $\mathbb{K} = \mathbb{F}_q$, up to reindexing the roots of P, the conjugates $\alpha_1, \ldots, \alpha_{d-1}$ of α_0 are just its iterated qth powers. Denoting $\mathbb{L} = \mathbb{K}[x]/(P(x)) = \mathbb{F}_{q^d}$, let us assume that $S(y) = y^2 - x$ is reducible in $\mathbb{L}[y]$, then $\beta_0 \in \mathbb{L}$. As such, one can choose $\beta_i = \beta_0^{q^i}$, the iterated qth powers. In that case, the previous equations can be rewritten

$$\begin{aligned} P(z) &= (z-\alpha_0)\left(z-\alpha_0^q\right)\cdots\left(z-\alpha_0^{q^{d-1}}\right) = (z-x)\left(z-x^q\right)\cdots\left(z-x^{q^{d-1}}\right), \\ Q(z) &= (z-\beta_0)\left(z-\beta_0^q\right)\cdots\left(z-\beta_0^{q^{d-1}}\right) = (z-y)\left(z-y^q\right)\cdots\left(z-y^{q^{d-1}}\right). \end{aligned}$$

As a consequence, $Q(z) \in \mathbb{K}[z]$ and to compute $Q(z)$, we need to compute y^{q^i} effectively. This is done by computing the following values in $O(d\log q)$ operations in $\mathbb{L}$:

$$u_0 = x, u_1 = x^q \bmod P(x), \ldots, u_{d-1} = u_{d-2}^q = x^{q^{d-1}} \bmod P(x).$$

Then, we simply compute in d operations $Q(z) = (z-u_0)(z-u_1)\cdots(z-u_{d-1})$ and we know that the resulting polynomial is in $\mathbb{K}[z]$.

Whenever α_0 is not a square in $\mathbb{L}$, that is whenever $S(y)$ is irreducible, then $\beta_0^{q^d}$ is a square root of α_0 different from β_0, thus it is $-\beta_0$. As a consequence, setting $Q(z) = (z-\beta_0)(z-\beta_0^q)\cdots(z-\beta_0^{q^{d-1}})$ would yield a polynomial that is not stable by the Frobenius endomorphism.

As such, we introduce a new variable y to represent the field $\mathbb{L}' = \mathbb{L}[y]/(y^2-x)$ and to compute $Q(z)$, we need to compute $y^{q^{i(d+1)}}$ effectively. Since $y^{q^i} = y y^{q^i-1} = y x^{\frac{q^i-1}{2}}$, we can compute the following values in $O(d\log q)$ field operations in $\mathbb{L}$:

$$u_0 = 1, u_1 = x^{\frac{q-1}{2}} \bmod P(x), \ldots, u_{d-1} = u_1\, u_{d-2}^q = x^{\frac{q^{d-1}-1}{2}} \bmod P(x).$$

Consequently, $Q(z) = (z - yu_0)(z - yu_1)\cdots(z - yu_{d-1})$.

As a first step, we compute in d operations, the dehomogenized polynomial in y,

$$\tilde{Q}(z) = (z-u_0)(z-u_1)\cdots(z-u_{d-1}) = z^d + h_1 z^{d-1} + \cdots + h_{d-1}z + h_d.$$

Then, $Q(z) = z^d + yh_1 z^{d-1} + \cdots + y^{d-1}h_{d-1}z + y^d h_d$. But, denoting by $i_0 = i \bmod 2$, we have $y^i = y^{i_0} y^{i-i_0} = y^{i_0} x^{\frac{i-i_0}{2}}$. Hence we deduce:

$$\begin{aligned} Q(z) &= z^d + yh_1 z^{d-1} + xh_2 z^{d-2} + yxh_3 z^{d-3} + \cdots + y^{d_0} x^{\frac{d-d_0}{2}} h_d \\ &= z^d + y \sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} h_{2i+1} x^i z^{d-2i-1} + \sum_{i=1}^{\lfloor \frac{d}{2} \rfloor} h_{2i} x^i z^{d-2i}. \end{aligned}$$

Complexity analysis. Since the number of operations for computing the square root of a block of size d is bounded by $O(d \log q)$ operations in $\mathbb{L} = \mathbb{F}_{q^d}$, this is also bounded by $O(d\mathsf{M}(d)\log q)$ operations in $\mathbb{K} = \mathbb{F}_q$, where $\mathsf{M}(n)$ is a bound on the number of operations in $\mathbb{K}$ to multiply two polynomials in $\mathbb{K}[x]$ of degree at most $n-1$. As a consequence, the computation of W can be done in no more than $O(n^\omega + n\mathsf{M}(n)\log q)$ operations in $\mathbb{K}$. Let us assume that the characteristic polynomial of Z has degree n and can be factored as $P_1^{e_1}\cdots P_s^{e_s}$ with P_i and P_j coprime whenever $i \neq j$, $\deg P_i = d_i$ and $e_i \geq 1$. From a computation point of view, in the worst case, one needs to introduce a variable α_i for one root of P_i and a variable β_i for the square root of α_i, assuming α_i is not a square. This yields a total number of $2s$ variables.

Computation in characteristic 2. The case of characteristic 2 is almost the same. From a polynomial $P(z) = z^d + p_{d-1}z^{d-1} + \cdots + p_0 = (z-\zeta_1)\cdots(z-\zeta_d)$, we want to compute $Q(z) = z^d + q_{d-1}z^{d-1} + \cdots + q_0 = (z-\omega_1)\cdots(z-\omega_d)$, with $\omega_i^2 = \zeta_i$ for all $1 \leq i \leq d$. As $P(z^2) = Q(z)^2$, this yields $q_i = \sqrt{p_i} = p_i^{q/2}$, for all $1 \leq i \leq d-1$. Thus, Q can be computed in $O(d\log q)$ operations in $\mathbb{K}$ and as a consequence, W in $O(n^\omega + n\log q)$ operations in $\mathbb{K}$.

However, let us recall that D is block diagonal if and only if the Jordan normal form is block diagonal. As such, a square root of D is a polynomial in D if and only if D is block diagonal, see Section 6.2.

Acknowledgements

We would like to thank Gabor IVANYOS for his helpful remarks and references on the irregular case. We wish to thank Gilles MACARIO-RAT for the many discussions about isomorphism of quadratic polynomials and Nitin SAXENA for those about graph isomorphism.

We thank the anonymous referees for their careful reading and their helpful comments.

This work has been partly supported by the French National Research Agency ANR-11-BS02-0013 HPAC project.

References

Agrawal, M., Saxena, N., 2006. Equivalence of F-Algebras and Cubic Forms. In: Durand, B., Thomas, W. (Eds.), STACS. Vol. 3884 of Lecture Notes in Computer Science. Springer, pp. 115–126.

Albrecht, M., Bard, G., 2012. The M4RI Library – Version 20121224. The M4RI Team. URL `http://m4ri.sagemath.org`

Bernardi, A., , Gimigliano, A. Idà, M., 2011. Computing symmetric rank for symmetric tensors. J. Symb. Comput. 46 (1), 34–53.

Berthomieu, J., Hivert, P., Mourtada, H., 2010. Computing Hironaka's invariants: Ridge and Directrix. In: Arithmetic, Geometry, Cryptography and Coding Theory 2009. Vol. 521 of Contemp. Math. Amer. Math. Soc., Providence, RI, pp. 9–20.

Bettale, L., Faugère, J.-C., Perret, L., 2013. Cryptanalysis of HFE, Multi-HFE and Variants for Odd and Even Characteristic. Designs, Codes and Cryptography 69 (1), 1 – 52.

Bhattacharyya, A., Fischer, E., Lovett, S., 2013. Testing low complexity affine-invariant properties. In: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1337–1355.

Bini, D., Pan, V. Y., 1994. Polynomial and Matrix Computations. Volume 1: Fundamental Algorithms. Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA.

Bosma, W., Cannon, J., Playoust, C., 1997. The Magma algebra system. I. The user language. J. Symbolic Comput. 24 (3-4), 235–265, computational algebra and number theory (London, 1993).

Bouillaguet, C., Faugère, J.-C., Fouque, P.-A., Perret, L., 2011. Practical cryptanalysis of the identification scheme based on the isomorphism of polynomial with one secret problem. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (Eds.), Public Key Cryptography. Vol. 6571 of Lecture Notes in Computer Science. Springer, pp. 473–493.

Bouillaguet, C., Fouque, P.-A., Véber, A., 2013. Graph-theoretic algorithms for the "isomorphism of polynomials" problem. In: Johansson, T., Nguyen, P. N. (Eds.), EUROCRYPT. Vol. 7881 of Lecture Notes in Computer Science. Springer, pp. 211–227.

Bürgisser, P., 2012. Prospects for geometric complexity theory. In: IEEE Conference on Computational Complexity. IEEE, p. 235.

Bürgisser, P., Ikenmeyer, C., 2011. Geometric complexity theory and tensor rank. In: Fortnow, L., Vadhan, S. P. (Eds.), STOC. ACM, pp. 509–518.

Bürgisser, P., Ikenmeyer, C., 2013. Explicit lower bounds via geometric complexity theory. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (Eds.), STOC. ACM, pp. 141–150.

Cai, J., 1994. Computing Jordan Normal forms Exactly for Commuting Matrices in Polynomial Time. International Journal of Foundations of Computer Science 05 (03n04), 293–302.

Carlini, E., 2005. Reducing the number of variables of a polynomial. In: Algebraic geometry and geometric modeling. Springer, pp. 237–247.

Carlitz, L., 03 1954. Representations by quadratic forms in a finite field. Duke Mathematical Journal 21 (1), 123–137.

Chen, X., Kayal, N., Wigderson, A., 2011. Partial derivatives in arithmetic complexity and beyond. Foundations and Trends in Theoretical Computer Science 6 (1-2), 1–138.

Chistov, A. L., Ivanyos, G., Karpinski, M., 1997. Polynomial time algorithms for modules over finite dimensional algebras. In: Char, B. W., Wang, P. S., Küchlin, W. (Eds.), ISSAC. ACM, pp. 68–74.

Cohn, H., Umans, C., 2013. Fast matrix multiplication using coherent configurations. In: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA. SIAM, pp. 1074–1086.

Comon, P., Golub, G. H., Lim, L.-H., Mourrain, B., 2008. Symmetric tensors and symmetric tensor rank. SIAM J. Matrix Analysis Applications 30 (3), 1254–1279.

de Seguins Pazzis, C., 2010. Invariance of simultaneous similarity and equivalence of matrices under extension of the ground field. Linear Algebra and its Applications 433 (3), 618 – 624.

DeMillo, R., Lipton, R., 1978. A probabilistic remark on algebraic program testing. Information Processing Letters 7 (4), 192–194.

Edmonds, J., 1967. Systems of distinct representatives and linear algebra. Journal of Research of the National Bureau of Standards 718 (4), 242 – 245.

Faugère, J.-C., Lin, D., Perret, L., Wang, T., 2012. On enumeration of polynomial equivalence classes and their application to MPKC. Finite Fields and Their Applications 18 (2), 283 – 302.

Faugère, J.-C., Perret, L., 2006. Polynomial Equivalence Problems: Algorithmic and Theoretical Aspects. In: Vaudenay, S. (Ed.), EUROCRYPT. Vol. 4004 of Lecture Notes in Computer Science. Springer, pp. 30–47.

Gantmacher, F., 1959. The Theory of Matrices, Vol. 1. Chelsea.

Giraud, J., 1972. Étude locale des singularités. U.E.R. Mathématique, Université Paris XI, Orsay, cours de 3ème cycle, 1971–1972, Publications Mathématiques d'Orsay, No. 26.

Green, B. J., Tao, T., 2009. The distribution of polynomials over finite fields, with applications to the gowers norms. Contributions to Discrete Mathematics 4 (2).

Grigorescu, E., Wimmer, K., Xie, N., 2013. Tight lower bounds for testing linear isomorphism. Electronic Colloquium on Computational Complexity (ECCC), 17.

Hironaka, H., 1970. Additive groups associated with points of a projective space. Ann. of Math. (2) 92, 327–334.

Kaltofen, E., 1989. Factorization of polynomials given by straight-line programs. In: Randomness and Computation. JAI Press, pp. 375–412.

Kayal, N., 2011. Efficient algorithms for some special cases of the polynomial equivalence problem. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, Philadelphia, PA, pp. 1409–1421.

Kayal, N., 2012. Affine projections of polynomials: extended abstract. In: Karloff, H. J., Pitassi, T. (Eds.), STOC. ACM, pp. 643–662.

Lang, S., 2002. Algebra, 3rd Edition. Vol. 211 of Graduate Texts in Mathematics. Springer-Verlag, New York.

Lidl, R., Niederreiter, H., 1997. Finite fields, 2nd Edition. Vol. 20 of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, with a foreword by P. M. Cohn.

Macario-Rat, G., Plût, J., Gilbert, H., 2013. New Insight into the Isomorphism of Polynomial Problem IP1S and Its Use in Cryptography. In: Sako, K., Sarkar, P. (Eds.), Advances in Cryptology - ASIACRYPT 2013. Vol. 8269 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 117–133.

Mackey, D. S., Mackey, N., Tisseur, F., 2005. Structured factorizations in scalar product spaces. SIAM J. Matrix Anal. Appl. 27 (3), 821–850.

Matsumoto, T., Imai, H., 1988. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Advances in Cryptology – EUROCRYPT 1988. Vol. 330 of LNCS. Springer–Verlag, pp. 419–453.

Matthews, K. R., 1992. A rational canonical form algorithm. Math. Bohemica 117, 315–324.

Mulmuley, K., 2012. The GCT program toward the *P* vs. *NP* problem. Commun. ACM 55 (6), 98–107.

Mulmuley, K., Sohoni, M. A., 2001. Geometric Complexity Theory I: An Approach to the P vs. NP and Related Problems. SIAM J. Comput. 31 (2), 496–526.

Newman, M., 1967. Two classical theorems on commuting matrices. J. Res. Nat. Bur. Standards Sect. B 71B, 69–71.

Patarin, J., 1996. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: Maurer, U. M. (Ed.), EUROCRYPT. Vol. 1070 of Lecture Notes in Computer Science. Springer, pp. 33–48.

Patarin, J., Goubin, L., Courtois, N., 1998. Improved algorithms for isomorphisms of polynomials. In: Nyberg, K. (Ed.), EUROCRYPT. Vol. 1403 of Lecture Notes in Computer Science. Springer, pp. 184–200.

Perret, L., 2004. On the computational complexity of some equivalence problems of polynomial systems of equations over finite fields. Electronic Colloquium on Computational Complexity (ECCC) 116.

Perret, L., 2005. A fast cryptanalysis of the isomorphism of polynomials with one secret problem. In: Cramer, R. (Ed.), EUROCRYPT. Vol. 3494 of Lecture Notes in Computer Science. Springer, pp. 354–370.

Saxena, N., 2006. Morphisms of Rings and Applications to Complexity. Ph.D. thesis, Indian Institute of Technology Kanpur.

Shanks, D., 1973. Five number-theoretic algorithms. In: Proceedings of the Second Manitoba Conference on Numerical Mathematics (Univ. Manitoba, Winnipeg, Man., 1972). Utilitas Math., Winnipeg, Man., pp. 51–70. Congressus Numerantium, No. VII.

Singla, P., 2010. On representations of general linear groups over principal ideal local rings of length two. Journal of Algebra 324 (9), 2543–2563.

Storjohann, A., 1998. An $O(n^3)$ algorithm for the Frobenius normal form. In: Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation. ISSAC '98. ACM, New York, NY, USA, pp. 101–105.

Strassen, V., 1969. Gaussian elimination is not optimal. Numer. Math. 13, 354–356.

Tang, S., Xu, L., 2012. Proxy signature scheme based on isomorphisms of polynomials. In: Xu, L., Bertino, E., Mu, Y. (Eds.), NSS. Vol. 7645 of Lecture Notes in Computer Science. Springer, pp. 113–125.

Tang, S., Xu, L., 2014. Towards provably secure proxy signature scheme based on isomorphisms of polynomials. Future Generation Computer Systems 30, 91 – 97, special Issue on Extreme Scale Parallel Architectures and Systems, Cryptography in Cloud Computing and Recent Advances in Parallel and Distributed Systems, ICPADS 2012.

Valiant, L. G., 1979. The complexity of computing the permanent. Theor. Comput. Sci. 8, 189–201.

von zur Gathen, J., Gerhard, J., 1999. Modern computer algebra. Cambridge University Press, New York.

Wallenborn, L. A., 2013. Berechnung des Hilbert Symbols, quadratische Form-Äquivalenz und Faktorisierung ganzer Zahlen. Master's thesis, Rheinische Friedrich-Wilhelms-Universität.

Wolf, C., Preneel, B., 2011. Equivalent keys in multivariate quadratic public key systems. Journal of Mathematical Cryptology 4 (4), 375–415.

Yang, G., Tang, S., Yang, L., 2011. A novel group signature scheme based on mpkc. In: Bao, F., Weng, J. (Eds.), ISPEC. Vol. 6672 of Lecture Notes in Computer Science. Springer, pp. 181–195.

Zippel, R., 1979. Probabilistic algorithms for sparse polynomials. In: Symbolic and algebraic computation (EUROSAM'79), Internat. Sympos. Vol. 72 of Lecture Notes in Computer Science. Springer Verlag, pp. 216–226.

Appendix A. Proofs of #IP1S

In this appendix, we shall prove the dimension of the centralizer of a matrix J, a Jordan normal form. This dimension, a consequence of (Singla, 2010, Lemma 4.11), is used in Section 4 to determine an upper bound on the counting problem of quadratic-IP1S. As stated by Singla, the proofs only involve matrix multiplications are given in order for the paper to be self-contained.

First, let us recall that the centralizer of J, a Jordan block of size s is the set of upper triangular Toeplitz matrices of size $s \times s$. Indeed, if X commutes with J, $XJ - JX$ is as such

$$XJ - JX = \begin{pmatrix} -x_{2,1} & x_{1,1}-x_{2,2} & \cdots & x_{1,n-1}-x_{2,n} \\ \vdots & \vdots & & \vdots \\ -x_{n,1} & x_{n-1,1}-x_{n,2} & \cdots & x_{n-1,n-1}-x_{n,n} \\ 0 & x_{n,1} & \cdots & x_{n,n-1} \end{pmatrix} = 0.$$

This small result is used in the following Lemma to determine the centralizer of a Jordan normal form.

Lemma 22. *Let J be a Jordan normal form. For $1 \leq i \leq r$, let us denote J_i the ith block of J and let us assume it is associated with eigenvalue ζ_i and it is of size s_i. Let $X = (X_{i,j})_{1 \leq i,j \leq r}$ be a block-matrix, with $X_{i,j} \in \mathbb{L}(\zeta_1, \ldots, \zeta_r)^{s_i \times s_j}$, that commutes with J. If $\zeta_i = \zeta_j$, then $X_{i,j}$ is an upper triangular Toeplitz matrix whose nonnecessary zero coefficients are the one on the first* $\min(s_i, s_j)$ *diagonals. Otherwise, $X_{i,j} = 0$.*

Proof. We assume that $r = 2$. If $XJ - JX = \begin{pmatrix} X_{1,1}J_1 - J_1X_{1,1} & X_{1,2}J_2 - J_1X_{1,2} \\ X_{2,1}J_1 - J_2X_{2,1} & X_{2,2}J_1 - J_1X_{2,2} \end{pmatrix} = 0$, then $X_{1,1}$ commutes with $J_1 = J_{\zeta_1, s_1}$ and $X_{2,2}$ with $J_2 = J_{\zeta_2, s_2}$. Thus they are upper triangular Toeplitz matrices.

From $X_{2,1}J_2 - J_1X_{2,2}$, one deduces that $(\zeta_1 - \zeta_2)x_{s_1+s_2,1} = 0$, hence either $\zeta_1 = \zeta_2$ or $x_{s_1+s_2,1} = 0$. If $\zeta_1 \neq \zeta_2$, then step by step, one has $X_{1,2} = 0$. Assuming $\zeta_1 = \zeta_2$, then step by step, one has $x_{s_1+i,1} = 0$ for $i > 1$ and since $x_{s_1+i+1,j+1} - x_{s_1+i,j} = 0$ for all i, j, one has in fact that $X_{1,2}$ is a upper

triangular Toeplitz matrix with potential nonzero coefficients on the first $\min(s_1, s_2)$ diagonals. The same argument applies to $X_{2,1}$.

The case $r > 2$ is an easy generalization of this result. □

From this lemma, we can deduce easily the dimension of the centralizer of a matrix.

Lemma 23. *Let $H \in \mathbb{K}^{n\times n}$ be a matrix and let J be its normal Jordan form. Assuming the blocks of J associated to ζ_i are $J_{\zeta_i,s_{i,1}}, \ldots, J_{\zeta_1,s_{i,d_1}}$ with $s_{i,1} \leq \cdots \leq s_{i,d_i}$ for $i, 1 \leq i \leq r$, then the centralizer of H is a $\mathbb{K}$-vector subspace of $\mathbb{K}^{n\times n}$ of dimension no more than $\sum_{1\leq i\leq r}\sum_{1\leq j\leq d_i}(2d_i - 2j + 1)s_{i,j}$.*

Proof. Let $\mathbb{L}$ be the smallest field over which J is defined. It is clear that the centralizer of H over $\mathbb{L}$, denoted $\mathscr{W}$, contains $\mathscr{C}(H) \otimes \mathbb{L}$. Hence, $\dim_{\mathbb{K}} \mathscr{C}(H) = \dim_{\mathbb{L}}(\mathscr{C}(H) \otimes \mathbb{L}) \leq \dim_{\mathbb{L}} \mathscr{W}$.

Now, let $X = (X_{i,j})_{1\leq i,j\leq d_1+\cdots+d_r} \in \mathscr{V}$. From Lemma 22, there are $\sum_{1\leq i\leq r}\sum_{1\leq j\leq d_i} s_{i,j}$ free parameters for the diagonal blocks of X and $2\sum_{1\leq i\leq r}\sum_{1\leq j<k\leq d_i}\min(s_{i,j}, s_{i,k}) = 2\sum_{1\leq i\leq r}\sum_{1\leq j\leq d_i}(d_i - j)s_{i,j}$ free parameters for the off-diagonal blocks of X. This concludes the proof. □

As a consequence, the number of invertible matrices in $\mathscr{C}(H)$ is bounded from above by

$$q^{\left(\sum_{1\leq i\leq r}\sum_{1\leq j\leq d_i}(2d_i - 2j+1)\,s_{i,j}\right)} - 1,$$

as stated in Corollary 16.